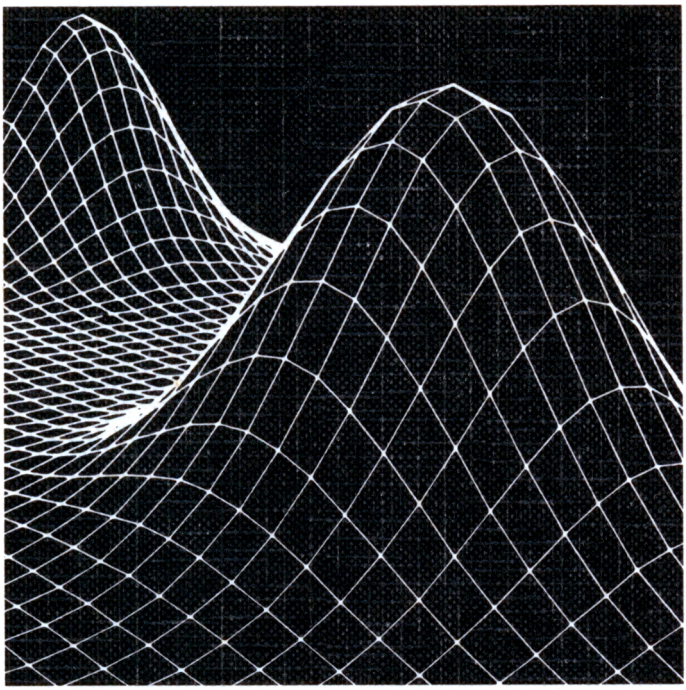
Markus Weber

3-D-Grafik in Theorie und Praxis

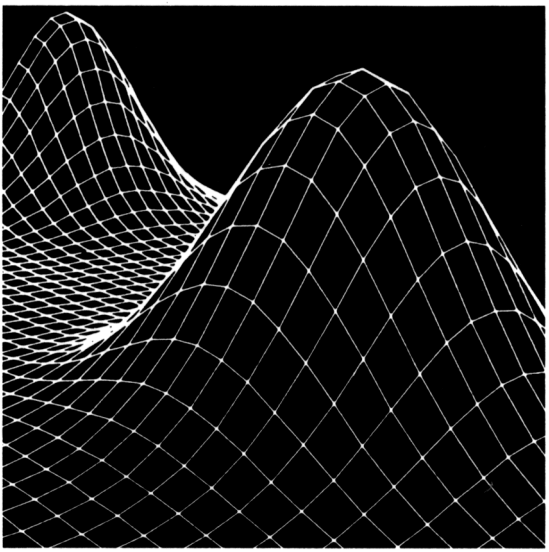


Hintergründe der Programmierung am Beispiel Commodore 64

iwt

Markus Weber

3-D-Grafik in Theorie und Praxis



Hintergründe der Programmierung am Beispiel **Commodore 64**

iwt

CIP-Kurztitelaufnahme der Deutschen Bibliothek

Weber, Markus:
3-D-Grafik [Drei-D-Grafik] in Theorie und Praxis:
Hintergründe d. Programmierung am Beispiel
Commodore 64 / Markus Weber. – Vaterstetten:
IWT-Verlag, 1984
ISBN 3-88322-052-3

Bildquellennachweis:
Max-Planck-Institut für Plasmaphysik und
Verlag Sterne und Weltraum Dr. Vehrenberg GmbH
Seite 11
Wir danken für die freundliche Genehmigung

ISBN 3-88322-052-3
1. Auflage 1984

Alle Rechte, auch die der Übersetzung, vorbehalten. Kein Teil des Werkes darf in irgendeiner Form (Druck, Fotokopie, Mikrofilm oder einem anderen Verfahren) ohne schriftliche Genehmigung des Verlages reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

Printed in Western Germany
© Copyright 1984 by IWT-Verlag GmbH
Vaterstetten bei München

Umschlag: Kaselow + Partner
Herstellung: Freiburger Graphische Betriebe, Freiburg

VORWORT

Dieser Band soll vor allem auch den mathematisch nicht übermäßig vorgebildeten Leser, ausgehend von einer allgemeinen Einführung in die Vektorrechnung über die Erstellung einfacher Grafiken zur Darstellung verschiedener Arten von dreidimensionalen Grafiken führen. Das Buch ist in zwei separate Abschnitte unterteilt, die jeder für sich selbständig gelesen werden können. Teil 2 setzt aber bereits Kenntnisse der analytischen Geometrie und der elementaren Matrizenrechnung voraus, die in Teil 1 behandelt werden.

Der erste Abschnitt des Buches beschäftigt sich ausschließlich mit den theoretischen Voraussetzungen für die Erstellung von dreidimensionalen Grafiken. Der Leser wird hier, ausgehend von einfachen bildhaften Beispielen, in den Umgang mit Vektoren eingeführt. In der Folge wird er mit elementaren Matrizenoperationen vertraut gemacht, die von enormer praktischer Bedeutung sind.

Im zweiten Abschnitt wird er anhand von einfachen Beispielprogrammen mit ersten Anwendungen der Vektoren in der Computergrafik vetraut gemacht. Darauf aufbauend schließen sich dann voll ausgebaute Programme aus den Bereichen Architekturperspektive, Systemsimulation, dreidimensionaler Analysis, Kugeldarstellungen und Kugelspiegelungen an.

Um den Leser nicht an das, für die Erstellung von Bildschirmgrafiken doch relativ langsame BASIC zu binden, findet sich im Anhang ein Maschinenprogramm, das durch ein BASIC-Programm eingelesen wird und im Zusammenwirken mit einer BASIC-Subroutine alle wesentlichen Grafikfunktionen ermöglicht. Der Übersichtlichkeit wegen wurde bei den in diesem Buch erläuterten Programmen auf das Grafiksystem IWT Sprite Komfort Kit zurückgegriffen. Ein Auszug aus dem Befehlslisting, der eine Umsetzung auf andere Grafiksysteme sowie ins BASIC ermöglicht, findet sich ebenfalls im Anhang.

Ich möchte an dieser Stelle den Herren Herrmann und Tröger für die mir entgegengebrachte

freundliche Unterstützung und den nötigen Anregungen zur rechten Zeit danken. Dem Verlag, insbesondere Herrn Grohmann, danke ich für die Herausgabe des Buches sowie die enorme Geduld, mit der er meine vielen Fragen immer bereitwillig beantwortet hat.

Forstinning, April 1984

Der Autor

INHALT

Vorwort.....3
Computergrafik - Warum ?.....9

TEIL 1

Vektorrechnung.....17
  Grundbegriffe.....17
  Vektor und Skalar.....17
  Addition und Multiplikation mit einem Skalar.....19

Der n-dimensionale Raum.....21

  Linearkombinationen von Vektoren.....21
  Koordinaten eines Vektors.....23
  Punktkoordinaten.....24
  Beispiele.....26

Geraden und Ebenen im R^3.....35

Anwendungen.....36

Multiplikation von Vektoren.....41

  Das Skalarprodukt.....41
  Eigenschaften.....42
  Das Vektorprodukt.....42
  Anwendungen von Skalar und Vektorprodukt.....45

Matrizen.....54

  Der Vektorraum.....54
  Basis und Dimension.....57
  Das Gauß'sche Eliminationsverfahren.....57
  Verknüpfungen von Matrizen.....63
  Lineare Abbildungen.....66
  Beispiele einfacher Matrizenoperationen.....75

TEIL 2

Aufbau des Grafikbildschirms.....85

Geometrische Grundoperationen im R^2 und R^3.....87

Verschiebung.....87
Spiegelungen.....88
Punktspiegelung.....88
Achsenspiegelung im R^2 und
Ebenenspiegelung im R^3.....89
Drehungen.....93

Grundstrukturen der Grafikprogrammierung.....95

Kreise.....95
Ellipsen.....96
Spiralen.....98

Schrägschrift.....101

Simulation des 3 Körperproblems.....103

Rotationsmatrizen.....111

Projektionstypen.....117

Zentralprojektion.....117
Parallelprojektion.....119
Anwendungen von Rotationsmatrix und
Zentralprojektion.....120

Das dreidimensionale Netz.....137

Zur Definition von dreidimensionalen
Funktionen.....153
Beispiele.....154
Elevation modell programming.....161

3D - Die zweite Version.....165

Dreidimensionale Kugeln.....171

Kugelspiegelungen.....177

Anhang A : Einfache Beispielprogramme der Computergrafik.....185

Anhang B : Planetensystem mit Grafik und Sprites.....190

Anhang C : Auszug aus dem Befehlslisting des IWT Sprite Komfort Kit.....194

Anhang D : Grafiksubroutinen (Maschinenhilfs-programm mit BASIC-Unterstützung)...195

Mit dem iwt-Programm auf die Zukunft programmiert!

Warum sich die Arbeit machen, die andere schon für Sie erledigt haben?

Lieferbare Programm-Disketten und -Kassetten zu den »Commodore 64«-Büchern aus dem IWT Verlag:

Mathematik auf dem Commodore 64
Kassette: Best.-Nr. 883 22 501 DM 58,–*
Diskette: Best.-Nr. 883 22 101 DM 58,–*

BASIC auf dem Commodore 64
Kassette: Best.-Nr. 882 22 501 DM 78,–*
Diskette: Best.-Nr. 882 22 101 DM 78,–*

Grafik auf dem Commodore 64
Kassette: Best.-Nr. 880 22 501 DM 58,–*
Diskette: Best.-Nr. 880 22 101 DM 58,–*

Wirtschaft auf dem Commodore 64
Kassette: Best.-Nr. 881 22 501 DM 58,–*
Diskette: Best.-Nr. 881 22 101 DM 58,–*

IWT Sprite Komfort Kit
Sammlung von Maschinen-Routinen für den Commodore 64

- 30 neue Befehle für das Arbeiten mit Sprites und der hochauflösenden Grafik
- superschnelle hochauflösende Grafik
- Tastatur-Wiederholfunktion
- Wandlung von Dezimal- in Hexa-Dezimalzahlen und umgekehrt
- Kreis- und Rahmenbefehle
- Abfrage von Disketten-Fehlern u. v. m.

Kassette: Best.-Nr. 850 22 501 DM 98,–
Diskette: Best.-Nr. 850 22 101 DM 98,–

* Gebundener Preis incl. MwSt. Preisänderungen vorbehalten.
Ihre Bestellung richten Sie bitte an:

IWT Software Service – für Information, Wissenschaft, Technologie
Altenberger Straße 23b, 5093 Burscheid, Tel. (02174) 62815, Tx 5213989 iwt
Verkaufsbüro: Dahlienstr. 4, 8011 Vaterstetten, Tel. (08106) 31017, Tx 5213989 iwt

IWT

'Ordnung ist das Reich der Logik,
Unordnung das der Phantasie'
(George Bernanos)

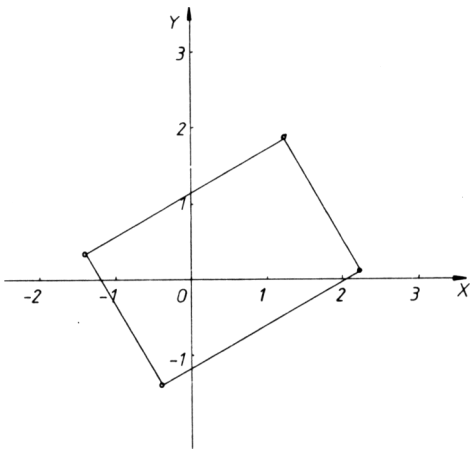
'Der Theoretiker kommt nach 4 Wochen
und meldet: Es ist mir jetzt
endlich gelungen, den Elefanten in
eine Kugel zu transformieren, jetzt
können wir anfangen zu rechnen. '
(Erich Sackmann)
Professor für Kernphysik und
nukleare Festkörperphysik an
der TU München

Computergrafik - Warum ?

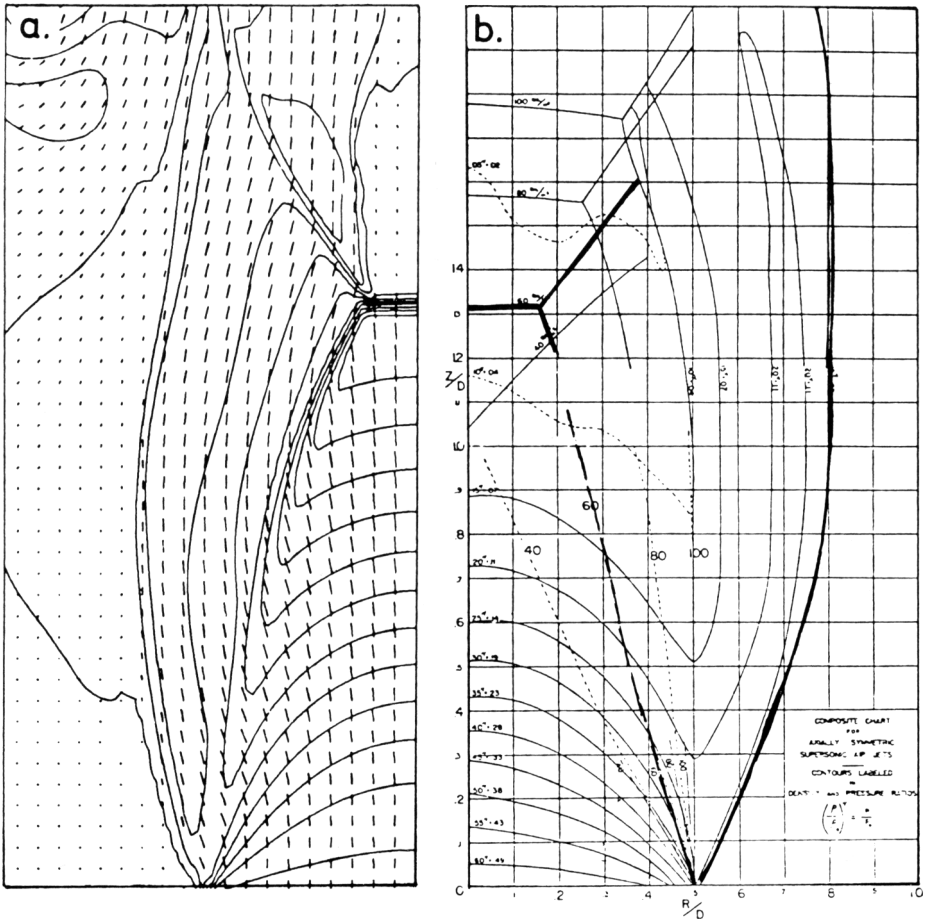
Dies ist ein Buch über Computergrafik. Aber wozu brauchen wir Computergrafiken überhaupt ? Diese Frage läßt sich eigentlich ganz einfach beantworten: Mit Hilfe von Computergrafiken werden dem Betrachter nüchterne mathematische Tatsachen plastisch vor Augen geführt. Wozu eigentlich? Wieder ist die Antwort ganz einfach: Was sagen Ihnen zum Beispiel die folgenden Zahlenpaare:

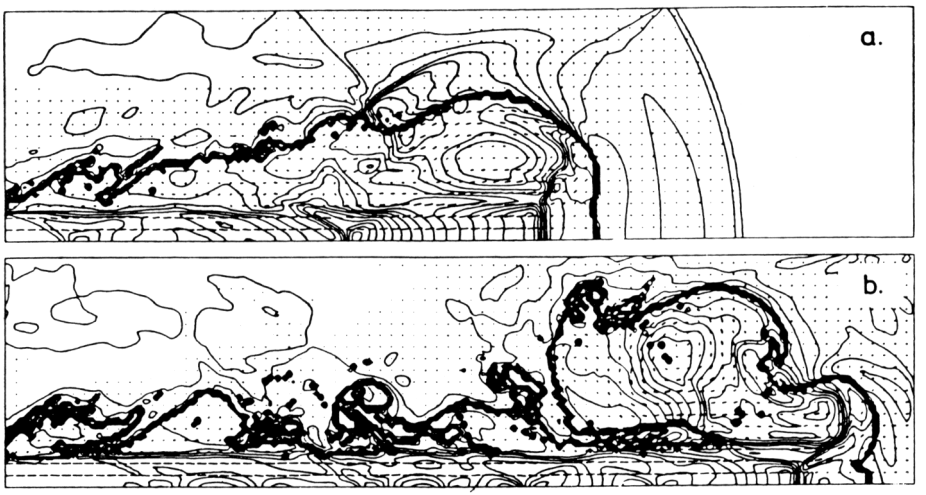
$$\begin{bmatrix}1.232\\1.866\end{bmatrix} \quad \begin{bmatrix}2.232\\0.134\end{bmatrix} \quad \begin{bmatrix}-1.366\\0.366\end{bmatrix} \quad \begin{bmatrix}-0.366\\-1.366\end{bmatrix}$$

Mit ziemlicher Sicherheit praktisch nichts.Diese Situation ändert sich aber schlagartig wenn wir diese Punkte in ein Koordinatensystem eintragen:



Offensichtlich ist das menschliche Gehirn nicht darauf eingestellt, digital verabreichte Information entsprechend zu verarbeiten. Wir sind darauf abgerichtet Informationen in erster Linie durch unser Auge, in zweiter Linie erst durch Gehör oder Tasten aufzunehmen und auch dementsprechend zu verarbeiten. Aber es heißt auch nicht umsonst 'Übung macht den Meister'. Der häufige Umgang mit digitalisierter Information führt dazu, daß sich manche Menschen so sehr daran gewöhnen, daß sie nicht zu weit gehende Informationen dieser Art gedanklich verarbeiten können. Doch das sind lediglich Einzelfalle, die die Regel bestätigen.Die breite Masse wird niemals dazu fähig sein.Die Schwierigkeiten, die sie dabei haben, werden alle diejenigen Leser zu spüren bekommen, die sich noch nie mit Vektorrechnung beschäftigt haben. Es wird Ihnen trotz Erklärungen und Skizzen erfahrungsgemäß anfangs sehr schwerfallen der Argumentation, vor allem im dreidimensionalen Raum zu folgen, da es einfach an der notwendigen Übung fehlt. Aber lassen sie sich davon nicht abschrecken; wir leben schließlich in einer dreidimensionalen Welt, und sie werden bald merken, wie sie mit ein bißchen Übung mit der Zeit der Argumentation leichter folgen können.
Der Computer selbst kann nicht dreidimensional rechnen, genaugenommen nicht einmal zweidimensional- er ist an Zahlenkolonnen gebunden. Er kann sie aber, mit den entsprechenden Programmen versehen, in dem Programmierer leicht verständliche Graphen umsetzen. Dies ist insofern wichtig, als mit fortschreitender Entwicklung immer größerer, leistungsfähigerer und nicht zuletzt schnellerer Computer und Software mehr und mehr Probleme mit dem Computer gelöst werden können. Der Computer erfreut sich nicht umsonst so großer Beliebtheit bei Industrie und Forschung. Mit seiner Hilfe lassen sich Experimente oft eingrenzen und wesentlich verbilligen, oder gar ganz ersetzen. Aus dem Bereich der Industrie sei hier die sog. 'Finite elemente'- Methode erwähnt. Dabei wird zum Beispiel ein Kraftfahrzeugpleuel im Computer in endlich viele Elemente zerlegt (= finite elemente) ,für die angenommen werden kann, daß sie so nahe an einen idealisierten Zustand herankommen, daß für sie bestimmte physikalische Gesetze, die sich z.B. theoretisch oder makroskopisch begründen lassen, wieder gelten. Normalerweise gehen solche Eigenschaften bei realen Körpern verloren. Für sehr kleine Elemente behalten sie jedoch meistens als sehr gute


a.
b.
Z/D
R/D
40
60
80
100
COMPOSITE CHART
FOR
AXIALLY SYMMETRIC
SUPERSONIC JET
CONTOURS LABELLED
IN
DENSITY AND PRESSURE RATIOS

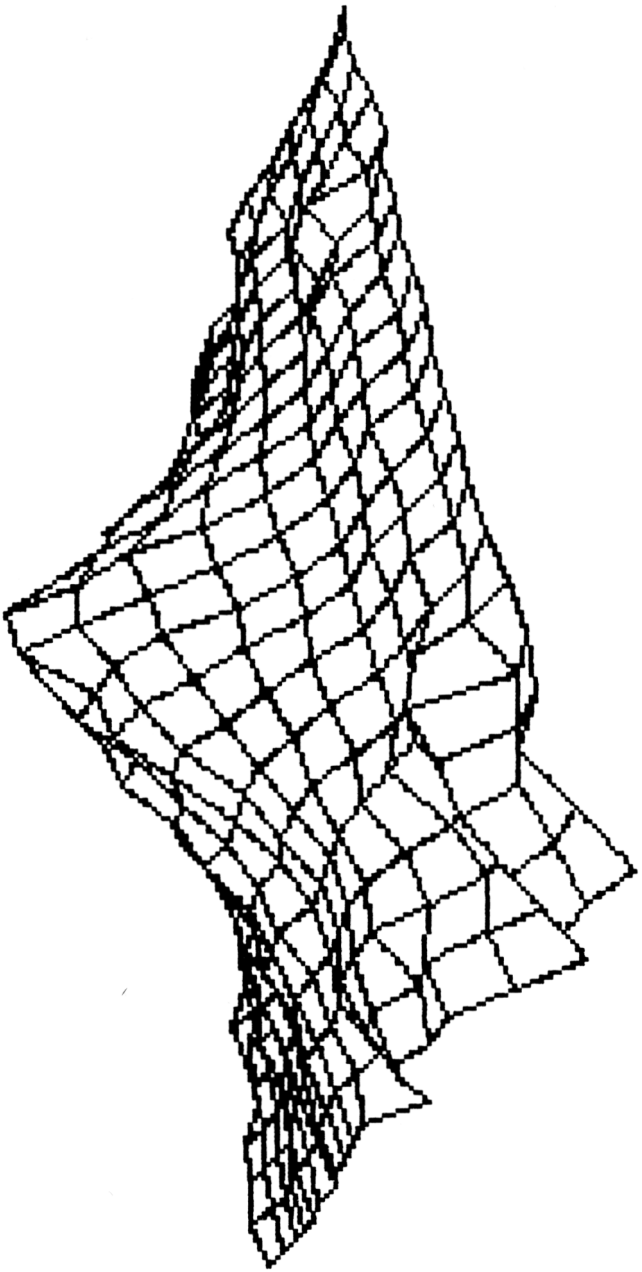

a.
b.

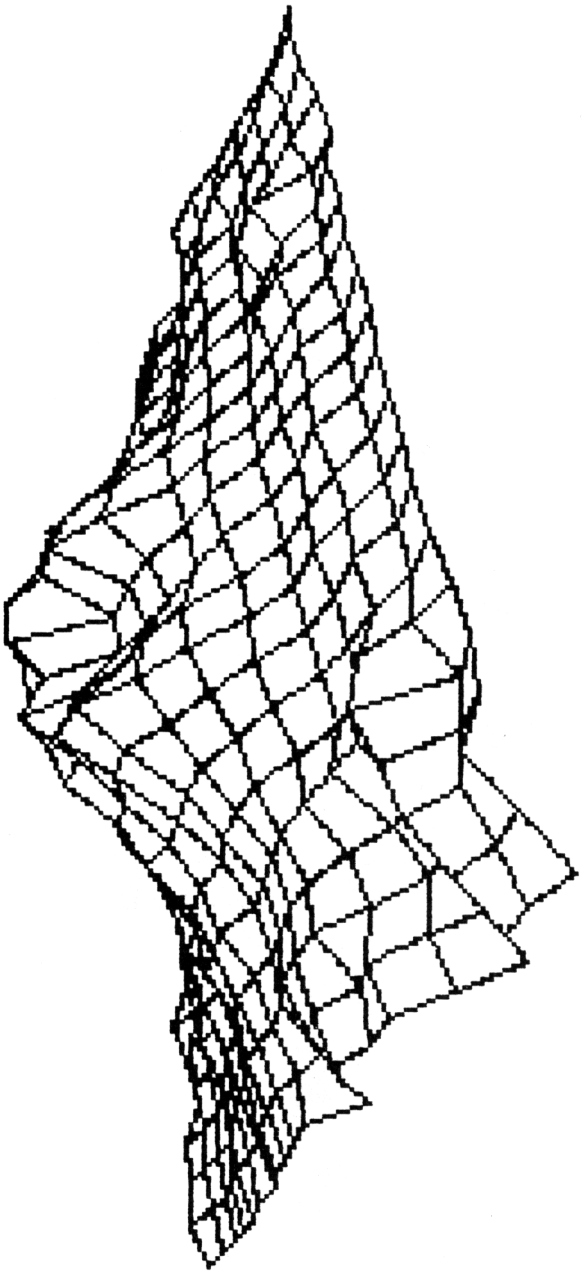
Näherung Gültigkeit. (Man vergleiche hierzu auch die Differentialgleichungen fester Körper, die letztenendes auf nichts anderem beruhen).

Mit Hilfe dieser Methode kann man das Pleuel dann im Computerexperiment verschiedenen Belastungen aussetzen und die Wirkung von Änderungen beobachten. Auch hier wird die Information hauptsächlich graphisch wiedergegeben.
Damit wären wir auch schon dort wo die Computergrafik ihr primäres Einsatzgebiet findet,nämlich bei der Verarbeitung von riesigen Datenmengen. Dabei lassen sich grundsätzlich zwei verschiedene Datenquellen unterscheiden: Interne und Externe.
Mit internen Datenquellen sind in erster Linie Computerexperimente gemeint, wie wir eben gesehen haben. Ein weiteres Beispiel hierfür ist z.B. die Anatomie eines Düsenstrahls direkt hinter der Austrittsöffnung, die wir in Abb.2 sehen. Die Computerechnung (a) stimmt dabei offensichlich sehr gut mit dem experimentell ermittelten Ergebnis überein.
Abbildung 3 zeigt zwei Stadien eines Jetkopfes beim Eindringen in ein Medium mit der zehnfachen Materiedichte, die er selbst besitzt. In diesem Computerexperiment entstehen Pilzwirbel am Kopfende des Überschall-Düsenstrahls, der mit niedrieger Dichte und Mach-3 Geschwindigkeit ausströmt. Die Rechnungen hierzu wurden 1980 von Michael Norman und Karl-Heinz Winkler vom Garchinger Max-Plank-Institut in Zusammenarbeit mit Larry Smarr und Michael Smith von der Universität von Illinois durchgeführt. Sie verwendeten dazu die CRAY-1, den damals schnellsten Computer der Welt. Statt sonst notwendiger 50 Stunden Rechenzeit genügten auf der CRAY-1 zwei bis drei Stunden. Angesichts der Tatsache, daß die CRAY-1 über 1 Million (!) Rechenoperationen in einer Sekunde ausführen kann, konnte man dabei praktisch nicht auf eine graphische Bearbeitung der Datenfluten durch den Computer verzichten.

Ein großes Anwendungsgebiet der Computergrafik ist die Bearbeitung externer Datenquellen, d.h. in erster Linie von Messdaten.

Bis zum 18.Mai 1980 galt der Mount St.Helens im US-Bundesstaat Washington als 'Bilderbuchvulkan'. Dies änderte sich jedoch, als an jenem Tag in einer aufsehenerregenden Eruption die oberen 400 Meter des Vulkans weggerissen und ein tiefer Krater ausgesprengt wurde. Die beiden





Computergrafiken zeigen den Berg vor und nach der Eruption jeweils aus nordöstlicher Richtung. Sie wurden vom Autor nach Daten des US Geological Survey angefertigt (vgl. 'Elevation modell programming' in Abschnitt 2 dieses Buches).

Ein weiteres Beipiel für den Einsatz der Computergrafik ist z.B. die Behandlung optischer Photographien vor allem in der Astronomie. Das menschliche Auge ist oft zu unzuverläßig, da es kleine Intensitäten neben großen nicht mehr zu erkennen vermag. Man verwendet hier das Verfahren des sog. 'Colour Coding'. Dabei wird jeder Intensität eine bestimmte Farbe zugeordnet und vorher unsichtbare Strukturen werden dank der wesentlich höheren Empfindlichkeit des Computers sichtbar. Was diese Methode besonders interessant macht ist die Tatsache , daß sie natürlich auch in Wellenlängenbereichen praktikabel ist, in denen das menschliche Auge blind ist, wie z.B. im Infraroten und im Bereich der Röntgen- oder Radiostrahlung. Eine Folge war die Entwicklung sog. CCD-Chips (charge-coupled devise), die die Photoplatte praktisch ersetzen. Ein CCD-Chip ist im Grunde genommen nichts anderes als ein überdimensionierter Silizium-Chip. Aber anstelle von Transistoren im Mikromaßstab, befinden sich darauf kleine lichtempfindliche Felder, sog. 'pixels' (für picture elements). Um es stark vereinfacht zu sagen: jedes Pixel ist ein Photonenzähler. Möglich wird dies durch einen Effekt den mittlerweile jeder kennt: die photovoltaischen Eigenschaften des Siliziums Photonen direkt in elektrische Ladung zu verwandeln. Im großen wird dieser Effekt zum Beispiel bei den Solarzellen für Satelliten eingesetzt.
Setzt man den CCD-Chip eine bestimmte Zeit einer Strahlung aus, so ist die in jedem Pixel aufgebaute Ladung (= charge) proportional zur auf ihn gefallenen Intensität. Nach dieser 'Belichtung' können diese Ladungen vom Computer in seinen Arbeitsspeicher ausgelesen werden.
Doch es ist nicht nur die digitale Ausgabeweise des CCD-Chips, die ihn so beliebt macht. Die neuesten CCD-Chips bringen es heute bereits auf 640.000 Pixels die aus 800 x 800 Spalten x Zeilen bestehen und 10-mal empfindlicher im blauen bis gelben Teil des Spektrums sind als die besten photographischen Platten. Dies ermöglicht es wesentlich kleinere Teleskope bei gleicher Leistung zu bauen. So ließe sich heute der 5-m Spiegel des Mount Palomar Teleskops, einem der größten Teleskope der Welt, leicht durch einen

0.4-m Spiegel, der mit einem CCD-Chip ausgerüstet ist, ersetzen.

Überhaupt ist die Astronomie ein beliebtes Einsatzgebiet für Computergrafiken. So lassen sich mit Hilfe des 'false-colour-coding' (Falschfarbencodierung) z.B. Geschwindigkeitsverteilungen in Galaxien ermitteln. Diese Methode beruht darauf, daß die von einem bewegten Teilchen ausgesandte Strahlung dem Dopplereffekt unterliegt. Dabei wird die Wellenlänge der Strahlung von Teilchen, die sich von der Erde entfernen länger, d.h. sie wird zum roten Ende des Spektrums hin verschoben, bzw. die von Teilchen, die sich der Erde nähern wird kürzer, sprich nach dem blauen Ende hin verschoben. Die auftretende Frequenzverschiebungen sind allerdings so gering, daß sie vom menschliche Auge nicht wahrgenommen werden können. Für den Computer dagegen ist es ein leichtes sie zu erkennen und dies in entsprechenden Graphiken wiederzugeben.

Wie sie sehen ist die Computergrafik heute bereits zu einem fast unverzichtbaren Hilfsmittel für Ingenieure und Naturwissenschaftler avanciert.

TEIL 1

Vektorrechnung

Ich werde im folgenden eine kurze Einführung in die Vektorrechnung geben, da diese Begriffe für das Verständnis der in diesem Buch erläuterten Programme unumgänglich sind. Ganz allgemein läßt sich aber auch sagen, daß die Vektorrechnung das vom mathematischen Aufwand und von der Anschaulichkeit her probateste Mittel ist schnelle und genaue Computergrafiken zu erzeugen. Wie wir sehen werden lassen sich aber bereits mit einem so 'kleinen' Computer wie dem Commodore 64 sehr nützliche und anspruchsvolle Grafiken erstellen. Es mag im ersten Augenblick so anmuten, daß ich hier ein Buch über Vektorrechnung und nicht über Computergrafik geschrieben habe.Dem ist aber keineswegs so. Auch wenn es zunaechst uneinsichtig erscheint: Die Mathematik ist das grundlegendste aller grafischen Hilfsmittel und nirgends sonst sehen wir die Wirkung von ein bißchen Mathematik so schnell wie bei Computergrafiken.
Bei den Programmen handelt es sich durchweg um solche,die sie durch leichte Modifikationen hervorragend für eigene Anwendungen verwenden können (z.B. zur Auswertung von Meßergebnissen). Ich habe dabei viel Wert auf die Möglichkeit der praktischen Anwendbarkeit (vor allem bei den 3D-Grafiken) gelegt, da es Aufgabe der Computergrafiken ist,neben der ihnen manchmal nicht abzusprechenden Ästhetik,in erster Linie unübersichtliche Zahlenkolonnen in schnell begreifbare,anschauliche Grafiken zu übersetzen.

Grundbegriffe

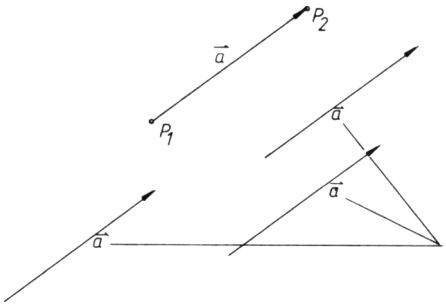
Vektor und Skalar

Wir müssen in der Mathematik zwischen zwei grundlegend verschiedenen Größen unterscheiden. Größen, deren Werte durch reelle Zahlen ausgedrückt werden können, wie z.B. Masse, Temperatur, Ladung, Längen etc., heißen SKALARE. Ihre Einführung und Charakterisierung ist Gegenstand der Analysis und ist das, was wir im allgemeinen mit 'Schulmathematik' bezeichnen. Größen dagegen, die durch eine Zahlenangabe und zusätzlich eine Richtung im Raum charakterisiert sind, wie z.B. Geschwindigkeit, Kraft, elektrische und magnetische Feldstärke etc., nennt man VEKTOREN. Im anschaulichen Sinn

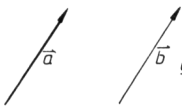
gesprochen ist ein Vektor eine gerichtete Strecke oder 'Pfeil' im Raum. (Notation: $\overline{P1P2}$;$\vec{a}$,$\vec{b}$,...;P1= Anfangspunkt,P2= Endpunkt). Die Länge der Strecke heißt BETRAG des Vektors $\vec{a}$ und wird durch $|\vec{a}|$ bezeichnet. Man unterscheidet sinnvollerweise zwischen zwei Gruppen von Vektoren; warum, werden wir später sehen:
EINHEITSVEKTOREN sind Vektoren der Länge (=Betrag des Vektors) 1. Der Nullvektor $\vec{0}$ ist ein Vektor, bei dem Anfangspunkt und Endpunkt zusammenfallen; sein Betrag ist Null, seine Richtung unbestimmt.

Zwei Vektoren werden als gleich bezeichnet wenn sie gleiche Länge und gleiche Richtung haben.

Die Menge aller Vektoren, die aus einem gegebenen Vektor durch Parallelverschiebung hervorgeht, d.h. alle zu $\vec{a}$ gleichen Vektoren, bezeichnet man als die zu $\vec{a}$ gehörigen FREIEN VEKTOREN. Einen Vektor der aus bestimmten Gründen nicht verschoben werden darf bezeichnet man als GEBUNDENEN VEKTOR oder ORTSVEKTOR.

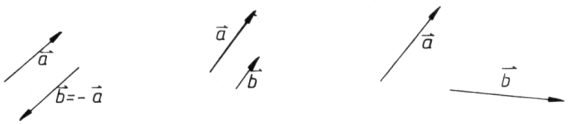


zu $\vec{a}$ gehörige freie Vektoren



gleiche Vektoren: $\vec{a}$ und $\vec{b}$ sind gleich lang und gleichgerichtet

Beispiele für ungleiche Vektoren

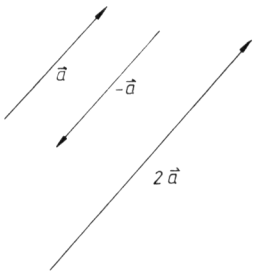


Die hier getroffene Definition eines Vektors ist

im mathematischen Sinn zu bildhaft und unzureichend. Wir werden hier auf diese Problematik nicht eingehen. Es sei nur erwähnt, daß sich z.B. auch Funktionen als Vektoren interpretieren lassen und dies auch in der höheren Analysis eine wichtige Rolle spielt. Wir werden uns jedoch auf Vektoren beschränken, für die die vorangegangene Definition vollkommen ausreichend ist.

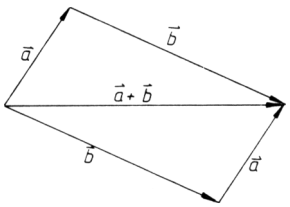
Addition und Multiplikation mit einem Skalar

Ist K eine reelle Zahl (in Zeichen K∈R; R = Menge der reellen Zahlen) und $\vec{a}$ ein Vektor, dann soll K$\vec{a}$ ein Vektor sein mit der Länge |K|*|$\vec{a}$| und der gleichen Richtung wie $\vec{a}$ im Falle K>0, und mit der entgegengesetzten Richtung im Falle K<0. Speziell hat also $-\vec{a}$ den gleichen Betrag wie $\vec{a}$ aber entgegengesetzte Richtung.Für den Fall K=0 definiert man Ka sinnvollerweise als $0*\vec{a} = \vec{o}$.

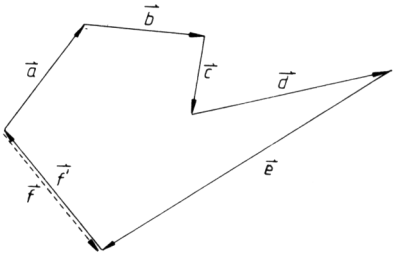


Durch Multiplikation eines Vektors mit einem Skalar soll sich also lediglich seine Länge, nicht aber seine Richtung ändern.

Sind $\vec{a}$ und $\vec{b}$ zwei Vektoren, dann wird ihre Summe wie folgt erklärt: Man setze nach geeigneter Parallelverschiebung $\vec{a}$ und $\vec{b}$ aneinander, d.h. den Anfangspunkt von $\vec{b}$ an den Endpunkt von $\vec{a}$. Der Summenvektor $\vec{a}+\vec{b}$ ist derjenige Vektor, der den gleichen Anfangspunkt wie $\vec{a}$ und den selben Endpunkt wie $\vec{b}$ besitzt. Wie aus untenstehender Abbildung ersichtlich, ist es egal ob man $\vec{b}$ an $\vec{a}$ oder $\vec{a}$ an $\vec{b}$ ansetzt, d.h. es gilt: $\vec{a}+\vec{b}=\vec{b}+\vec{a}$. Diese Eigenschaft bezeichnet man als Kommutativität.



Das Hintereinandersetzen, sprich die Addition, mehrer Vektoren bezeichnet man als Polygonzug.



Allgemein definiert man dann die Summe $\vec{a}+\vec{b}+...+\vec{n}$ mehrerer Vektoren $\vec{a},\vec{b},..,\vec{n}$ durch denjenigen Vektor $\vec{f}$, der dem Vektor $\vec{f}'$, der den Polygonzug schließt, entgegengesetzt ist:

$$\vec{a}+\vec{b}+...+\vec{n}+\vec{f}' = \vec{0}$$
$$\Leftrightarrow \vec{a}+\vec{b}+...+\vec{n} = -\vec{f}' = \vec{f}$$

Die Differenz $\vec{a}-\vec{b}$ wird erklärt durch die Summe aus $\vec{a}$ und $(-\vec{b})$:

$$\vec{a}-\vec{b} = \vec{a}+(-\vec{b})$$

Allgemein gelten die folgenden Rechengesetze:

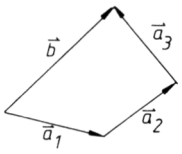
$\vec{a}+\vec{b} = \vec{b}+\vec{a}$		Kommutativität
$\vec{a}+(\vec{b}+\vec{c}) = (\vec{a}+\vec{b})+\vec{c}$		Assoziativität
$K(l\vec{a}) = Kl(\vec{a})$		$K,l \in R$
$K(\vec{a}+\vec{b}) = K\vec{a}+K\vec{b}$		Distributivität
$(K+l)\vec{a} = K\vec{a} + l\vec{a}$		$K,l \in R$
$\lvert K\vec{a}\rvert = \lvert K\rvert\lvert\vec{a}\rvert$		

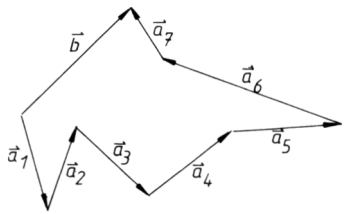
Offensichtlich gelten alle Gesetze wie wir sie aus der Analysis für reelle Zahlen Kennen, auch für Vektoren.

Der n-dimensionale Raum

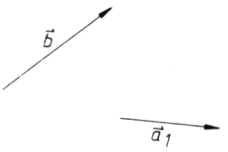
LinearKombinationen von VeKtoren

Um den Begriff des n-dimensionalen Raumes verstehen zu Können bedarf es einiger wichtiger Eigenschaften von VeKtoren. Eine ganz elementare Eigenschaft ist die LinearKombination:
Unter einer LinearKombination von VeKtoren a1,a2,a3,...,an versteht man den VeKtor b=K1a1+K2a2+...+Knan, wobei Ki, 1<=i<=n reelle Koeffizienten sind.
Wir veranschaulichen uns diese Tatsache an einem Beispiel:





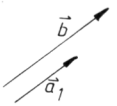
In beiden Fällen entsteht der VeKtor b durch LinearKombinationen von ai.Offensichtlich gibt es unendlich viele MöglichKeiten den VeKtor b durch verschiedene ai auszudrücKen.
Interessant ist hier in erster Linie die Frage ob es eine minimale Anzahl von ai gibt mit denen sich alle beliebigen VeKtoren b darstellen lassen?
Betrachten wir das folgende Beispiel:



Wir Können $\overline{b}$ offensichtlich nicht in der Form

$$\overline{b} = K\overline{a1}$$

darstellen, denn durch einen SKalar K wird die Richtung von $\overline{a1}$ nicht beeinflusst.



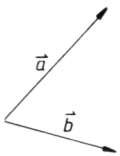
In diesem Beispiel__ Können wir $\overline{b}$__ als LinearKombination von $\overline{a1}$ darstellen, denn $\overline{a1}$ und $\overline{b}$ unterscheiden sich lediglich in ihrem Betrag, nicht aber in ihrer Richtung. Es gilt also:

$$\overline{b} = K\overline{a1} \qquad ,K \in R$$

Allgemein bezeichnet man zwei VeKtoren $\overline{a1}$ und $\overline{a2}$ als linear abhängig wenn gilt:

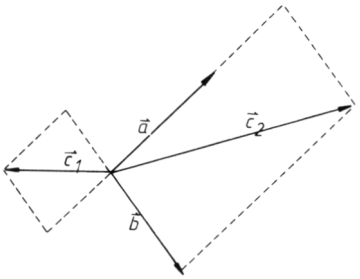
$$K1\overline{a1}+K2\overline{a2} = \overline{o} \qquad (*)$$

Zwei VeKtoren $\overline{a1}$ und $\overline{a2}$ heißen linear unabhängig wenn sich die Gleichung (*) nur trivial, d.h.K1=K2=0 lösen läßt.
Betrachten wir uns jetzt einen zweidimensionalen Raum, z.B. die Zeichenebene:



Die VeKtoren $\overline{a}$ und $\overline{b}$ sind linear unabhängig, da nur die Koeffizienten K1=K2=0 die Gleichung (*) lösen.
Es stellt sich nun die Frage ob sich noch weitere

Vektoren, z.B. ein Vektor $\overline{c}$, finden lassen, die ebenfalls in der Zeichenebene von $\overline{a}$ und $\overline{b}$ liegen, von diesen aber linear unabhängig sind.
Es ist anschaulich sofort klar daß dies unmöglich ist.



Wir haben also gefunden daß sich bereits durch zwei linear unabhängige Vektoren alle anderen Vektoren der selben Ebene darstellen lassen. Damit gilt allgemein folgender Satz:

Die Dimension des Raumes ist gleich der maximal möglichen Anzahl der linear unabhängigen Vektoren aus ihm.

Auf den genauen mathematischen Beweis gehen wir hier nicht ein. Dieser ist aber nicht besonders schwer zu führen.

Koordinaten eines Vektors

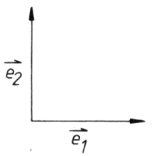
Wir beschränken uns im folgenden weiter auf den zweidimensionalen Raum (kurz R^2 ;lies R zwei), alles Gesagte gilt aber in gleicher Form auch für den R^n.
Seien $\overline{e1}$ und $\overline{e2}$ linear unabhängige Vektoren des R^2. Dann lässt sich jeder Vektor $\overline{b}$ des R^2 in der Form

$$\overline{b}=b1\overline{e1}+b2\overline{e2} \qquad ;b1,b2\in R$$

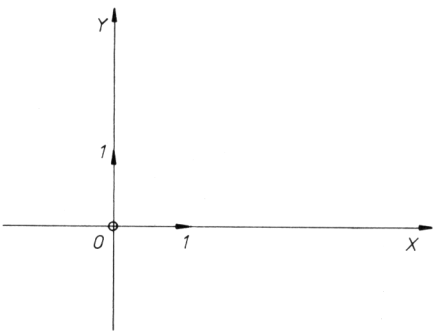
darstellen.
Bei gegebenen $\overline{e1},\overline{e2}$ wird die Richtung von $\overline{b}$ nur durch die Koeffizienten $\overline{b1}$ und $\overline{b2}$ bestimmt. Diese heißen die Koordinaten des Vektors $\overline{b}$ bzgl. der Basis $(\overline{e1},\overline{e2})$. Man schreibt allgemein kurz:

$$\overline{b} = \begin{bmatrix} b1 \\ b2 \end{bmatrix} \quad \text{oder} \quad \overline{b} = (b1/b2)$$

Als Basis bezeichnet man eine minimale Menge von linear unabhängigen Vektoren ej, durch die sich alle Vektoren des betrachteten Raumes eindeutig darstellen lassen.
Eine spezielle Basis ist die folgende:



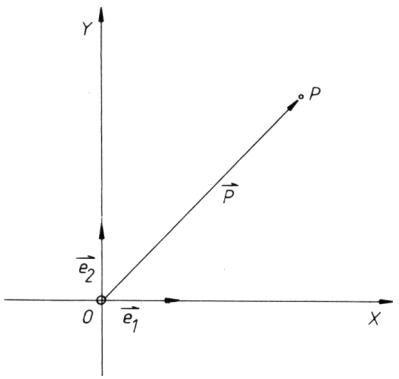
$\overline{e1}$ und $\overline{e2}$ sind Einheitsvektoren die aufeinander senkrecht stehen.Eine solche Basis heißt orthonormierte (=senkrechte und genormte) Basis oder Standardbasis.
Betrachten wir z.B. das x-y-Achsenkreuz. Die beiden Achsen entstehen durch Multiplikation der Einheitsvektoren mit allen reelle Zahlen R.



Punktkoordinaten

Wir können jetzt den Zusammenhang zwischen den Koordinaten eines Vektors und den Koordinaten eines Punktes herstellen. Trägt man wie oben in einem festen Punkt O zwei (bzw. drei,...,n) paarweise aufeinander senkrechte (orthogonale) Einheitsvektoren an, dann entsteht ein rechtwinkliges Koordinatensystem.

Wir können nun jedem Punkt P des Koordinatensystems einen Vektor $\overline{OP}$ zuordnen. Diesen Vektor bezeichnet man als Ortsvektor $\overline{P}$ von P.



Man sieht jetzt sofort den Zusammenhang zwischen den Koordinaten (x,y) eines Punktes $\overline{P}$ und seiner Vektordarstellung

$$\overline{P} = \begin{bmatrix} x \\ y \end{bmatrix} .$$

Der Ortsvektor läßt sich anhand der Punktkoordinaten eindeutig ermitteln, denn es gilt:

$$\overline{P} = x\overline{e1} + y\overline{e2}$$

Es ist also völlig ausreichend die Koordinaten eines Punktes zu Kennen, wenn von der Basis nur bekannt ist daß es sich um eine Standardbasis handelt.
Wie man jetzt leicht einsehen kann müssen die Basisvektoren $\overline{b1}, \overline{b2}$ gar nicht senkrecht aufeinander stehen oder Einheitsvektoren sein. Es genügt daß sie linear unabhängig sind. Wir werden auf diesen Sachverhalt später noch einmal zurückkommen.

Beispiele:

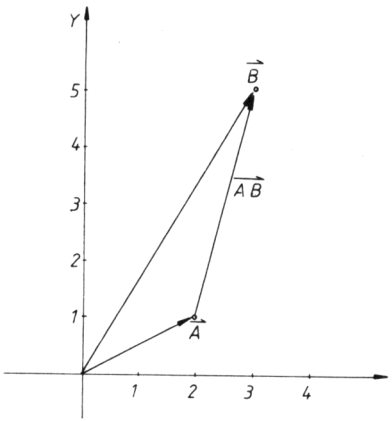
1)Die Basisvektoren $\overline{e1}$ und $\overline{e2}$ haben in Koordinatenschreibweise die Darstellung:

$$\overline{e1} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \; ; \; \overline{e2} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Es gilt ja

$$\overline{e1} = 1\overline{e1} + 0\overline{e2} \quad \text{bzw.} \quad \overline{e2} = 0\overline{e1} + 1\overline{e2}$$

2)Wir betrachten die Ortsvektoren $\overline{A} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$ und $\overline{B} = \begin{bmatrix} 3 \\ 5 \end{bmatrix}$



Wie können wir den Vektor bestimmen der die Punkte A und B verbindet,also den Vektor $\overline{AB}$?
Wir lösen das Problem anhand eines geschlossenen Polygonzuges:
Es muss gelten:

$$\overline{A} + \overline{AB} - \overline{B} = \overline{o}$$
$$\overline{A} + \overline{AB} = \overline{B} \qquad / \text{ Addition von B auf beiden Seiten}$$
$$\overline{AB} + \overline{A} = \overline{B} \qquad / \text{ Kommutativgesetz}$$
$$\overline{AB} = \overline{B} - \overline{A} \qquad / \text{ Subtraktion von A auf beiden Seiten}$$

Wir haben damit eine Formel für den Verbindungsvektor zweier Punkte gefunden und es gilt:

$$\overline{AB} = \overline{B}-\overline{A}$$

(Man beachte,daß gilt Verbindungsvektor = Endpunkt - Anfangspunkt und nicht wie die Schreibweise $\overline{AB}$ suggerieren könnte $\overline{A}-\overline{B}$)

Wir wollen nun noch die Koordinatennotation von $\overline{AB}$ ermitteln.
Wir definieren dazu:

$$\overline{a} = \begin{bmatrix} a1 \\ a2 \end{bmatrix} \quad ; \quad \overline{b} = \begin{bmatrix} b1 \\ b2 \end{bmatrix}$$

$$K\overline{a} = K \begin{bmatrix} a1 \\ a2 \end{bmatrix} = \begin{bmatrix} Ka1 \\ Ka2 \end{bmatrix}$$

$$\overline{a}+\overline{b} = \begin{bmatrix} a1 \\ a2 \end{bmatrix} + \begin{bmatrix} b1 \\ b2 \end{bmatrix} = \begin{bmatrix} a1+b1 \\ a2+b2 \end{bmatrix}$$

Allgemein: Haben $\overline{a}$ und $\overline{b}$ die Darstellungen

$$\overline{a} = \begin{bmatrix} a1 \\ a2 \\ . \\ . \\ an \end{bmatrix} \quad ; \quad \overline{b} = \begin{bmatrix} b1 \\ b2 \\ . \\ . \\ bn \end{bmatrix}$$

dann ist

$$\overline{a}+\overline{b} = \begin{bmatrix} a1+b1 \\ a2+b2 \\ . \\ . \\ an+bn \end{bmatrix} \quad ; \quad K\overline{a} = \begin{bmatrix} Ka1 \\ Ka2 \\ . \\ . \\ Kan \end{bmatrix}$$

Wir erhalten also für $\overline{AB}$

$$\overline{AB} = \overline{B}-\overline{A} = \overline{B}+(-\overline{A}) = \begin{bmatrix} 3 \\ 5 \end{bmatrix} +(-1) \begin{bmatrix} 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 3 \\ 5 \end{bmatrix} + \begin{bmatrix} -2 \\ -1 \end{bmatrix} = \begin{bmatrix} 3-2 \\ 5-1 \end{bmatrix} = \begin{bmatrix} 1 \\ 4 \end{bmatrix}$$

3)Wir untersuchen die Vektoren

$$\begin{bmatrix} 1 \\ 3 \\ 4 \end{bmatrix}, \begin{bmatrix} 2 \\ 6 \\ 8 \end{bmatrix}, \begin{bmatrix} 3 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 9 \\ 11 \end{bmatrix}$$

auf linesre Abhängigkeit.

Wie man unschwer erkennt gilt:

$$2\begin{bmatrix}1\\3\\4\end{bmatrix}=\begin{bmatrix}2\\6\\8\end{bmatrix}$$

Die ersten beiden Vektoren unterscheiden sich also lediglich in ihrer Länge, nicht aber in ihrer Richtung.
Etwas schwieriger wird es mit den beiden anderen Vektoren.Wie man anhand der ersten Koordinate des letzten Vektors erkennt, läßt er sich nicht durch einen der beiden Vektoren

$$\begin{bmatrix}1\\3\\4\end{bmatrix}\quad\text{oder}\quad\begin{bmatrix}3\\0\\1\end{bmatrix}$$

allein darstellen, denn es müßte gelten:

$$k\begin{bmatrix}1\\3\\4\end{bmatrix}=\begin{bmatrix}0\\9\\11\end{bmatrix}$$

<=>(wegen erster Koordinate) k=0

$$\Rightarrow 0\begin{bmatrix}1\\3\\4\end{bmatrix}=\begin{bmatrix}0\\0\\0\end{bmatrix}=\begin{bmatrix}0\\9\\11\end{bmatrix}\quad\text{Widerspruch !}$$

bzw.

$$k\begin{bmatrix}3\\0\\1\end{bmatrix}=\begin{bmatrix}0\\9\\11\end{bmatrix}$$

<=>(wegen erster Koordinate) k=0

$$\Rightarrow 0\begin{bmatrix}3\\0\\1\end{bmatrix}=\begin{bmatrix}0\\0\\0\end{bmatrix}=\begin{bmatrix}0\\9\\11\end{bmatrix}\quad\text{Widerspruch !}$$

Analoge Überlegungen gelten für den Vektor

$$\begin{bmatrix}3\\0\\1\end{bmatrix}.$$

Es bleibt zu prüfen ob sich z.B der Vektor

$$\begin{bmatrix}0\\9\\11\end{bmatrix}$$

als Linearkombination von $\begin{bmatrix}1\\3\\4\end{bmatrix}$ und $\begin{bmatrix}3\\0\\1\end{bmatrix}$ darstellen läßt.

Es müßte gelten :

$$K1\begin{bmatrix}1\\3\\4\end{bmatrix} + K2\begin{bmatrix}3\\0\\1\end{bmatrix} = \begin{bmatrix}0\\9\\11\end{bmatrix}$$

Es ergeben sich also 3 Teilgleichungen,wobei alle durch zwei eindeutig bestimmbare Werte K1 und K2 erfüllt werden müssen:

$$\begin{array}{lrcrcr} (1) & K1 & + & 3\,K2 & = & 0 \\ (2) & 3\,K1 & + & & = & 9 \\ (3) & 4\,K1 & + & K2 & = & 11 \end{array}$$

Aus (2) erhält man: K1 = 3
Einsetzen in Gleichung (1) liefert: 3+ 3K2 = 0
=> K2 = -1

PROBE:Einsetzen in Gleichung (3)

4*3-1 = 12-1 = 11

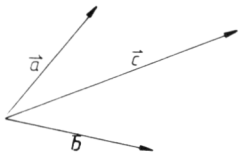
Die Werte K1=3 und K2=-1 sind also die Lösung des Gleichungssystems. Damit gilt:

$$3\begin{bmatrix}1\\3\\4\end{bmatrix} - \begin{bmatrix}3\\0\\1\end{bmatrix} = \begin{bmatrix}0\\9\\11\end{bmatrix}$$

Der Vektor $\begin{bmatrix}0\\9\\11\end{bmatrix}$ ist also von von den beiden

Vektoren $\begin{bmatrix}1\\3\\4\end{bmatrix}$ und $\begin{bmatrix}3\\0\\1\end{bmatrix}$ linear abhängig.

Wir lernen jetzt noch eine andere Methode kennen Vektoren auf lineare Abhängigkeit zu untersuchen. Diese Methode empfiehlt sich jedoch nur wenn man wissen will ob die Vektoren linear abhängig sind, nicht aber wie. Hierzu ist eine Vorbemerkung notwendig:
Wir betrachten 3 Vektoren $\bar{a},\bar{b},\bar{c}$ der Zeichenebene:



Wie wir bereits festgestellt haben ist es im zweidimensionalen Raum nicht möglich mehr als zwei linear unabhängige Vektoren zu finden, denn ein dritter läßt sich immer in der Form

$$\overline{c} = k\overline{a} + l\overline{b}$$

darstellen.
Dies läßt sich verallgemeinern: Im n-dimensionalen Raum lassen sich höchstens n linear unabhängige Vektoren finden, und jeweils n+1 Vektoren sind linear abhängig.
Hat man also 4 Vektoren des R^3, wie in unserem Beispiel, dann kann man von vornherein sagen, daß mindestens einer linear abhängig von den drei anderen ist. Es ist also jeweils nur notwendig je 3 Vektoren auf lineare Abhängigkeit zu untersuchen. Dies kann man anhand einer sogenannten Determinate tun. (Wir gehen hier nur auf zwei- bzw. dreireihige Determinanten ein.)

Es seien

$$\overline{a2} = \begin{bmatrix} a1 \\ a2 \end{bmatrix}, \overline{b2} = \begin{bmatrix} b1 \\ b2 \end{bmatrix}, \overline{a3} = \begin{bmatrix} a1 \\ a2 \\ a3 \end{bmatrix}, \overline{b3} = \begin{bmatrix} b1 \\ b2 \\ b3 \end{bmatrix}$$

$$\overline{c3} = \begin{bmatrix} c1 \\ c2 \\ c3 \end{bmatrix}$$

Vektoren des R^2 bzw. R^3.
Dann ist definiert:

$$\det \begin{vmatrix} a1 & b1 \\ a2 & b2 \end{vmatrix} = a1b2-a2b1$$

$$\det \begin{vmatrix} a1 & b1 & c1 \\ a2 & b2 & c2 \\ a3 & b3 & c3 \end{vmatrix} =$$

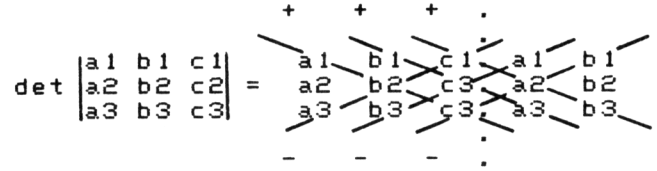
$$= a1b2c3+b1c2a3+c1a2b3-a3b2c1-b3c2a1-c3a2b1$$

Eine Merkregel für dreireihige Determinanten ist die sogenannte Regel von Sarrus:

Zu bestimmen ist

$$\det \begin{vmatrix} a1 & b1 & c1 \\ a2 & b2 & c2 \\ a3 & b3 & c3 \end{vmatrix}$$

Dann gilt:



Man addiert die Produkte der durch die Linien verbundenen, mit einem '+' versehenen, Koeffizienten und subtrahiert die Produkte der mit einem '-' versehenen.

Wie untersucht man nun mit Hilfe einer Determinante die lineare Abhängigkeit von Vektoren?
Hier gilt der Satz:

det A = 0 <=> Vektoren sind linear abhängig
det A # 0 <=> Vektoren sind linear unabhängig

$$\text{mit } A = \begin{bmatrix} a1 & b1 & c1 \\ a2 & b2 & c2 \\ a3 & b3 & c3 \end{bmatrix}$$

Wollen wir also wissen ob

$$\begin{bmatrix} 1 \\ 3 \\ 4 \end{bmatrix}, \begin{bmatrix} 3 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 9 \\ 11 \end{bmatrix}$$ linear abhängig sind,so bestimmt man

$$\det \begin{vmatrix} 1 & 3 & 0 \\ 3 & 0 & 9 \\ 4 & 1 & 11 \end{vmatrix} =$$

$$= 0+(3*9*4)+0-0-(1*9*1)-(11*3*3)=108-9-99 =$$

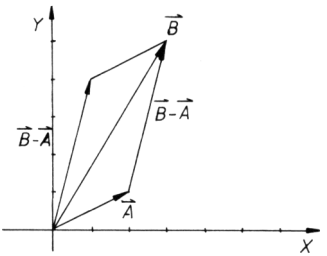
$$= 108-108 = 0$$

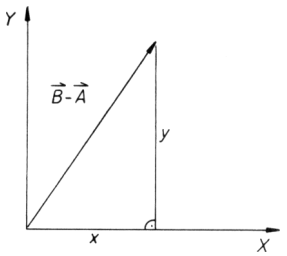
Die Vektoren sind also linear abhängig.
(Wichtig: Ist n die Dimension des Raumes, so können nur n Vektoren mit Hilfe der Determinante

auf lineare Abhängigkeit untersucht werden, nicht mehr und auch nicht weniger. Also 2 im R^2, 3 im R^3 usw.)

4) Wir können den Abstand der Punkte $\overline{A}$ und $\overline{B}$ aus (2) wie folgt bestimmen:

Voraussetzung ist ein orthonormiertes Koordinatensystem:





In (2) haben wir $\overline{B}-\overline{A} = \begin{bmatrix} 1 \\ 4 \end{bmatrix}$ bestimmt.

Nach dem Lehrsatz des Pythagoras gilt:

$$x^2 + y^2 = (|\overline{B}-\overline{A}|)^2 \iff |B-A| = \sqrt{x^2 + y^2}$$

Weiter ist:

$|\overline{B}-\overline{A}| = |\overline{AB}|$ = Abstand der Punkte A und B

Wir kommen also zu der Formel für den Betrag oder die Länge eines Vektors in einem orthonormierten Koordinatensystem.

$$|\overline{AB}| = |\overline{B}-\overline{A}| = \sqrt{x^2 + y^2}$$

wobei $\overline{AB}$ die Darstellung $\overline{AB} = \begin{bmatrix} x \\ y \end{bmatrix}$ besitzt.

da $x = b1-a1$
$y = b2-a2$

gilt:

$$|\overline{AB}| = \sqrt{(b1-a1)^2 + (b2-a2)^2}$$

Der Abstand der Punkte beträgt also $\sqrt{16+1} = \sqrt{17}$

5) Oftmals ist es notwendig den zu einem Vektor gehörigen Einheitsvektor zu bestimmen.Für ihn gilt:

$$\overline{a0} = \frac{\overline{a}}{|a|}$$

In Koordinaten also

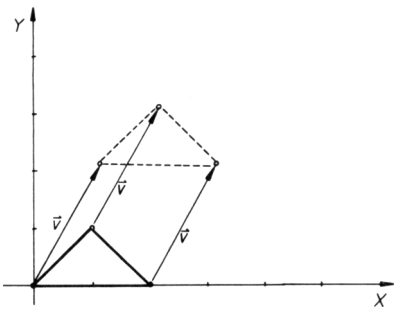
$$\overline{a0} = \frac{\begin{bmatrix} a1 \\ a2 \\ . \\ . \\ an \end{bmatrix}}{\sqrt{a1^2 + a2^2 + \ldots + an^2}}$$

Was haben wir nun damit gewonnen? Grob gesprochen die Möglichkeit mit Punkten oder geometrischen Gebilden zu rechnen, oder um es anders auszudrücken, wir können jetzt die uns zur Verfügung stehenden algebraischen und analytischen Mittel für die Geometrie verwenden. Dies führte zu dem Ausdruck 'Analytische Geometrie', die erstmals von Descartes systematisch behandelt wurde.

Wir können nun z.B. mit Hilfe eines Vektors das Dreieck

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 2 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

an eine beliebige Position verschieben.



Sei $\overline{v} = \begin{bmatrix} v1 \\ v2 \end{bmatrix}$

der Verschiebungsvektor für den Punkt (0/0), dann erhalten wir die neuen Koordinaten durch Addition von $\overline{v}$ zu den Ortskoordinaten des Dreiecks:

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \begin{bmatrix} 2 \\ 0 \end{bmatrix} \quad \begin{bmatrix} 1 \\ 1 \end{bmatrix} \longrightarrow \begin{bmatrix} v1 \\ v2 \end{bmatrix} \quad \begin{bmatrix} 2+v1 \\ v2 \end{bmatrix} \quad \begin{bmatrix} 1+v1 \\ 1+v2 \end{bmatrix}$$

6) Die Vektoren $\begin{bmatrix} 1 \\ 2 \end{bmatrix}, \begin{bmatrix} 2 \\ 1 \end{bmatrix}$ sind eine Basis des R^2.

BEWEIS: Offensichtlich sind beide Vektoren linear unabhängig. Die Standardeinheitsvektoren $\overline{e1}$ und $\overline{e2}$ lassen sich wie folgt darstellen:

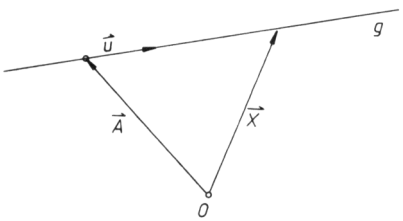
$$\overline{e1} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} = -\frac{1}{3}\begin{bmatrix} 1 \\ 2 \end{bmatrix} - \frac{2}{3}\begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

$$\overline{e2} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \frac{2}{3}\begin{bmatrix} 1 \\ 2 \end{bmatrix} - \frac{1}{3}\begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

Da sich $\overline{e1}$ und $\overline{e2}$ durch unsere vermeintlichen Basisvektoren darstellen lassen, durch $\overline{e1}$ und $\overline{e2}$ aber alle Punkte des R^2 darstellbar sind, sind auch durch $\begin{bmatrix} 1 \\ 2 \end{bmatrix}$ und $\begin{bmatrix} 2 \\ 1 \end{bmatrix}$ alle Punkte des R^2 darstellbar.

Geraden und Ebenen im R^3

Wir beschränken uns im folgenden auf den R^3, da höherdimensionale Fälle selten vorkommen und sich der R^2 als Spezialfall des R^3 von selbst ergibt. Betrachten wir untenstehende Abbildung



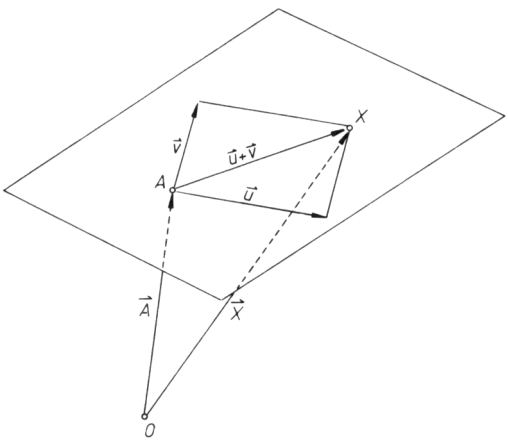
Es seien der Ursprung O,die Gerade g sowie ein Punkt A der Geraden g und ihr Richtungsvektor $\overline{u}$ gegeben.
Wir können die Punkte der Geraden wie folgt beschreiben:

$$g: \overline{X} = \overline{A}+k\overline{u} \quad ,k \in R \quad (*)$$

Denn der Vektor $\overline{X}$ schließt den Polygonzug

$$\overline{A}+k\overline{u}-\overline{X} = \overline{A}+k\overline{u}-\overline{A}-k\overline{u} = \overline{o}$$

Die Gleichung (*) bezeichnet man als Parameterdarstellung einer Geraden.
Zur Darstellung einer Ebene in Parameterform betrachten wir die folgende Abbildung:



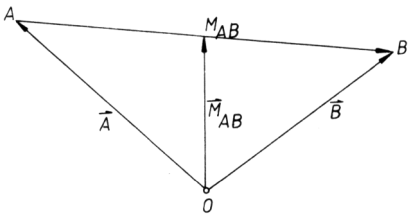
Wie wir bereits gesehen haben genügen 2 linear unabhängige Vektoren um alle Punkte des R^2 darzustellen. Verschieben wir den R^2 um einen Vektor $\overline{A}$ im Raum, so erhalten wir die allgemeine Darstellung einer Ebene in Parameterform.

$$E: \overline{X} = \overline{A} + K\overline{U} + l\overline{V} \quad ; K, l \in R$$

Anwendungen:

Wir wollen mit dem bisher erworbenen Wissen nun einige Anwendungen zum besseren Verständnis betrachten.

(1) Mittelpunkt einer Strecke $\overline{AB}$



Der Punkt $\overline{M}ab$ soll $\overline{AB}$ im Verhältnis 1:1 teilen. Es gilt also:

$$\overline{AM}ab = \overline{MabB}$$

bzw. $$|\overline{AM}ab| = |\overline{MabB}|$$

Wir bestimmen zunächst die Gleichung der Geraden die durch die Punkte A und B geht:
Wir wählen A als Trägerpunkt und verwenden $\overline{AB}$ als Richtungsvektor.

$$g: \overline{X} = \overline{A} + K(\overline{B} - \overline{A})$$

Der Punkt $\overline{M}ab$ liegt auf g. Der ihm zugeordnete Parameterwert K beträgt 0.5, da er $\overline{AB}$ halbiert.
Es gilt also:

$$\overline{M}ab = \overline{A} + \frac{1}{2}(\overline{B} - \overline{A}) = \overline{A} + \frac{1}{2}\overline{B} - (\frac{1}{2})\overline{A} = \frac{A+B}{2}$$

(2) Schwerpunkt eines Dreiecks

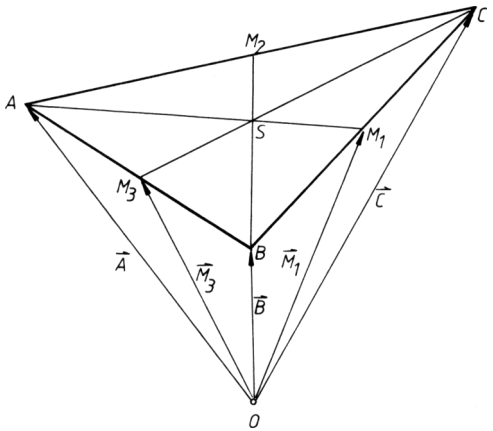
Aus der Geometrie wissen wir, daß der Schwerpunkt eines Dreiecks im Schnittpunkt der

Seitenhalbierenden liegt.
Wir wählen einen Punkt O als Ursprung, der nicht in der durch A,B und C bestimmten Ebene liegt. Dann haben die Ortsvektoren der Seitenmittelpunkte die Darstellung:

$$\overline{M1} = \frac{1}{2}(\overline{B}+\overline{C})$$

$$\overline{M2} = \frac{1}{2}(\overline{C}+\overline{A})$$

$$\overline{M3} = \frac{1}{2}(\overline{A}+\overline{B})$$



Die Seitenhalbierenden haben die Parameterdarstellung:

$$\overline{AM1}: \overline{X} = \overline{A}+r\left(\frac{1}{2}\overline{B}+\frac{1}{2}\overline{C}-\overline{A}\right) = (1-r)\overline{A}+\frac{r}{2}\overline{B}+\frac{r}{2}\overline{C}$$

$$\overline{BM2}: \overline{X} = \overline{B}+s\left(\frac{1}{2}\overline{C}+\frac{1}{2}\overline{A}-\overline{B}\right) = \frac{s}{2}\overline{A}+(1-s)\overline{B}+\frac{s}{2}\overline{C}$$

$$\overline{CM3}: \overline{X} = \overline{C}+t\left(\frac{1}{2}\overline{A}+\frac{1}{2}\overline{B}-\overline{C}\right) = \frac{t}{2}\overline{A}+\frac{t}{2}\overline{B}+(1-t)\overline{C}$$

Für den Schnittpunkt $\overline{S}$ von $\overline{AM1}$ und $\overline{BM2}$ gilt dann:

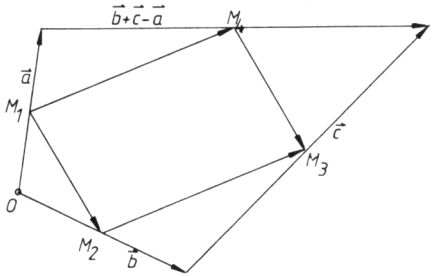
$$(1-r)\overline{A}+\frac{r}{2}\overline{B}+\frac{r}{2}\overline{C} = \frac{s}{2}\overline{A}+(1-s)\overline{B}+\frac{s}{2}\overline{C}$$

$$\Leftrightarrow \left(1-r-\left(\frac{s}{2}\right)\right)\overline{A}+\left(\left(\frac{r}{2}\right)-1+s\right)\overline{B}+\left(\left(\frac{r}{2}\right)-\left(\frac{s}{2}\right)\right)\overline{C}=\overline{0}$$

Da die Vektoren $\overline{A},\overline{B},\overline{C}$ nach Voraussetzung und wegen der Wahl von O linear unabhängig sind müßen die Koeffizienten von $\overline{A},\overline{B},\overline{C}$ jeweils 0 sein. Damit erhält man r=s=2/3.
Berechnet man analog den Schnittpunkt von $\overline{BM2}$ und $\overline{CM3}$ so erhält man s=t=2/3. Es ergibt sich in beiden Fällen derselbe Punkt $\overline{S}$. Die drei Seitenhalbierenden schneiden sich in einem Punkt. $\overline{S}$ heißt der Schwerpunkt des Dreiecks und hat den Ortsvektor

$$\overline{S} = \frac{\overline{A}+\overline{B}+\overline{C}}{3}$$

(3) Die Seitenmittelpunkte eines jeden Vierecks (im R^2) bilden ein Parallelogramm



Wir legen zweckmäßigerweise einen Eckpunkt in den Ursprung O.
Wir erhalten die Gleichungen für die Seitenmittelpunkte Mi:

$$\overline{M1} = \frac{1}{2}\overline{a}$$

$$\overline{M2} = \frac{1}{2}\overline{b}$$

$$\overline{M3} = \vec{b} + \frac{1}{2}\vec{c}$$

$$\overline{M4} = \vec{a} + \frac{1}{2}(\vec{b} + \vec{c} - \vec{a})$$

Für die Verbindungsvektoren $\overline{MiMj}$ erhalten wir dann:

$$\overline{M1M2} = \frac{1}{2}\vec{b} - (\frac{1}{2})\vec{a} = \frac{1}{2}(\vec{b} - \vec{a})$$

$$\overline{M4M3} = \vec{b} + \frac{1}{2}\vec{c} - \vec{a} - (\frac{1}{2})(\vec{b} + \vec{c} - \vec{a}) = \vec{b} + \frac{1}{2}\vec{c} - \vec{a} - (\frac{1}{2})\vec{b}$$

$$-(\frac{1}{2})\vec{c} + \frac{1}{2}\vec{a} = \frac{1}{2}\vec{b} - (\frac{1}{2})\vec{a} = \frac{1}{2}(\vec{b} - \vec{a})$$

$$\overline{M1M4} = \vec{a} + \frac{1}{2}(\vec{b} + \vec{c} - \vec{a}) - (\frac{1}{2})\vec{a} = \frac{1}{2}\vec{a} + \frac{1}{2}\vec{b} + \frac{1}{2}\vec{c} - (\frac{1}{2})\vec{a} =$$

$$= \frac{1}{2}\vec{b} + \frac{1}{2}\vec{c} = \frac{1}{2}(\vec{b} + \vec{c})$$

$$\overline{M2M3} = \vec{b} + \frac{1}{2}\vec{c} - (\frac{1}{2})\vec{b} = \frac{1}{2}\vec{b} + \frac{1}{2}\vec{c} = \frac{1}{2}(\vec{b} + \vec{c})$$

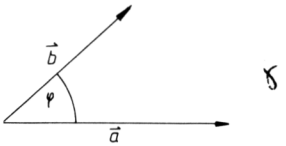
Die gegenüberliegenden Seiten $\overline{M1M2}$, $\overline{M4M3}$ und $\overline{M1M4}$, $\overline{M2M3}$ sind also jeweils parallel und gleichlang und spannen damit ein Parallelogramm auf. Q.E.D.

Multiplikation von Vektoren

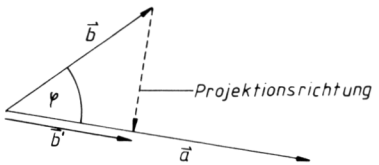
Wir haben uns bisher lediglich mit rein additiven Eigenschaften der Vektoren beschäftigt. Weitere Probleme ergeben sich bei der Messung oder Bestimmung von Abständen und Winkeln. Da Vektoren gerichtete 'Pfeile' sind,lassen sich auch Winkel zwischen ihnen ermitteln.

Das Skalarprodukt

Wir definieren das Skalarprodukt wie folgt:
Unter dem skalaren Produkt der Vektoren $\overline{a}$ und $\overline{b}$ (Schreibweise $\overline{ab}$, $\overline{a}o\overline{b}$, $\langle \overline{a},\overline{b} \rangle$) versteht man die reelle Zahl $\overline{ab} := |\overline{a}||\overline{b}|\cos\varphi$, wobei φ ($0 <= \varphi <= \pi$) der von $\overline{a}$ und $\overline{b}$ eingeschlossene spitze Winkel ist.



Wichtig:Das Ergebnis des Skalarprodukts ist KEIN Vektor sondern eine reelle Zahl.
Anschaulich läßt sich das Skalarprodukt wie folgt deuten: Sind $\overline{a}$ und $\overline{b}$ Vektoren des R^2 oder R^3, so ist das Skalarprodukt das Produkt des Betrages des Vektors $\overline{a}$ mit dem Betrag des in $\overline{a}$ projezierten Vektors $\overline{b}$.



Für die Koordinatenschreibweise ist $\overline{a}o\overline{b}$ wie folgt definiert.
Sei

$$\overline{a} = \begin{bmatrix} a1 \\ a2 \\ . \\ . \\ an \end{bmatrix}, \quad \overline{b} = \begin{bmatrix} b1 \\ b2 \\ . \\ . \\ bn \end{bmatrix}$$

dann ist

$$\vec{a}\circ\vec{b} = \begin{bmatrix} a1 \\ a2 \\ . \\ . \\ an \end{bmatrix} \circ \begin{bmatrix} b1 \\ b2 \\ . \\ . \\ bn \end{bmatrix} = a1b1+\ldots+anbn = \sum_{i=1}^{n} aibi$$

Eigenschaften des Skalarproduktes

S1: $(\vec{v}+\vec{u})\circ\vec{w} = \vec{v}\vec{w}+\vec{u}\vec{w}$
$\vec{v}\circ(\vec{u}+\vec{w}) = \vec{v}\vec{u}+\vec{v}\vec{w}$ Distributivität

S2: $\vec{v}\circ(k\vec{u}) = k(\vec{v}\circ\vec{u})$
$(k\vec{v})\circ\vec{w} = k(\vec{v}\circ\vec{w})$ Assoziativität (bzgl. der Multiplikation mit einer reellen Zahl)

S3: $\vec{v}\circ\vec{u} = \vec{u}\circ\vec{v}$ Kommutativität bzw. Symmetrie

S4: $\vec{v}\circ\vec{v} = \vec{v}^2 > 0$ für $v \neq o$
$\vec{v}^2 = |\vec{v}|^2$

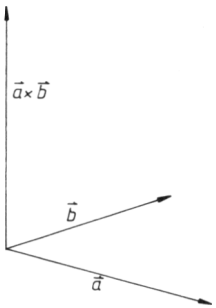
Das Skalarprodukt weißt also im wesentlichen alle Eigenschaften des algebraischen Produkts zweier Zahlen auf. Dementsprechend leicht ist auch seine Verwendung.
Wichtig:Definitionsgemäß ist folgender Ausdruck nicht moeglich:

$$\vec{a}\circ\vec{b}\circ\vec{c} = ?$$

da $(\vec{a}\circ\vec{b})\circ\vec{c} = \vec{a}\circ\vec{b}\circ\vec{c}$ gelten müßte, aber $\vec{a}\circ\vec{b} \in R$, d.h. eine richtungslose Zahl wäre. Da man jedoch keinen Winkel zwischen der Zahl $(\vec{a}\circ\vec{b})$ und dem Vektor $\vec{c}$ bestimmen kann ist auch der Ausdruck $\vec{a}\circ\vec{b}\circ\vec{c}$ nicht möglich.

Das Vektorprodukt (nur im R^3 definiert!)

Neben dem Skalarprodukt definiert man noch ein weiteres Produkt, das Vektor- oder Kreuzprodukt. Im Unterschied zum Skalarprodukt ist das Ergebnis des Kreuzprodukts wieder ein Vektor, der senkrecht auf der von $\vec{a}$ und $\vec{b}$ aufgespannten Ebene steht, und zwar so, daß $\vec{a}$, $\vec{b}$, $(\vec{a}\times\vec{b})$ ein rechtshändiges System bilden.



Unter einem rechtshändigen System versteht man ein System in dem, wenn man Daumen,Zeige- und Mittelfinger der rechten Hand gegeneinander rechtwinklig spreizt so daß der Daumen in Richtung von $\overline{a}$ und der Zeigefinger in Richtung von $\overline{b}$ zeigen, der Mittelfinger die Richtung von $\overline{a}x\overline{b}$ angibt (Rechte-Hand Regel).
Für die Koordinatenschreibweise des Vektorprodukts $\overline{a}x\overline{b}$ gilt folgende Merkregel:

Seien $\overline{a} = \begin{bmatrix} a1 \\ a2 \\ a3 \end{bmatrix}$ und $\overline{b} = \begin{bmatrix} b1 \\ b2 \\ b3 \end{bmatrix}$ Vektoren des R^3.

Dann ist

$$\begin{bmatrix} a1 \\ a2 \\ a3 \end{bmatrix} \times \begin{bmatrix} b1 \\ b2 \\ b3 \end{bmatrix} = \begin{bmatrix} \det \begin{matrix} a2 & b2 \\ a3 & b3 \end{matrix} \\ -\det \begin{matrix} a1 & b1 \\ a3 & b3 \end{matrix} \\ \det \begin{matrix} a1 & b1 \\ a2 & b2 \end{matrix} \end{bmatrix} =$$

$$= \begin{bmatrix} a2b3-a3b2 \\ a3b1-a1b3 \\ a1b2-a2b1 \end{bmatrix}$$

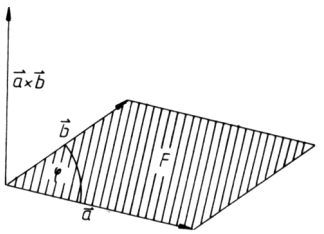
Man erhält also die i-te Koordinate, indem man die i-ten Koordinaten von $\overline{a}$ und $\overline{b}$ streicht und die zweireihige Restdeterminante berechnet. Wichtig ist, daß man vor die 2-te Koordinate ein Minuszeichen setzt.

Für das Vektorprodukt gelten die folgenden

Eigenschaften:

V1: $\vec{a}\times\vec{b} = -(\vec{b}\times\vec{a})$ (nicht Kommutativ !)
V2: $(k\vec{a})\times\vec{b} = k(\vec{a}\times\vec{b})$
V3: $\vec{a}\times(\vec{b}+\vec{c}) = \vec{a}\times\vec{b}+\vec{a}\times\vec{c}$

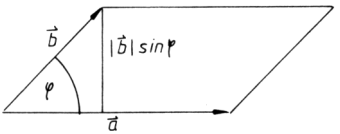
Anschaulich geprochen gibt das Vektorprodukt den Flächeninhalt des von $\vec{a}$ und $\vec{b}$ aufgespannten Parallelogramms wieder:



$|\vec{a}\times\vec{b}| = F$

Begründung:

$F = |\vec{a}|\,|\vec{b}|\sin\varphi =: \vec{a}\times\vec{b}$



Wichtig sind vor allem zwei Spezialfälle von Skalar- und Vektorprodukt:

$\vec{a}\circ\vec{b} = 0$,falls $\vec{a}\perp\vec{b}$,da dann $\cos\varphi = \cos 90° = 0$ gilt
$\vec{a}\times\vec{b} = o$,falls $\vec{a}\parallel\vec{b}$,da dann $\sin\varphi = \sin 0° = 0$ gilt

Häufig sind auch folgende Sätze von Nutzen:

(1) $(\vec{a}\times\vec{b})\vec{c} = \vec{a}(\vec{b}\times\vec{c})$
(2) $\vec{a}\times(\vec{b}\times\vec{c}) = (\vec{a}\vec{c})\vec{b}-(\vec{a}\vec{b})\vec{c}$ (Grassmann-Identität)
(3) $\vec{a}\times(\vec{b}\times\vec{c})+\vec{b}\times(\vec{c}\times\vec{a})+\vec{c}\times(\vec{a}\times\vec{b}) = \vec{o}$

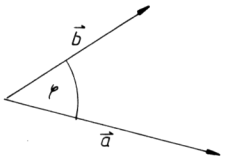
Anwendungen von Skalar- und Vektorprodukt

(1) Winkel zwischen zwei Vektoren

Es gilt:

$$\vec{a} \circ \vec{b} = |\vec{a}|\,|\vec{b}| \cos\phi$$

$$\Leftrightarrow \cos\phi = \frac{\vec{a} \circ \vec{b}}{|\vec{a}|\,|\vec{b}|}$$



Anhand dieser Formel läßt sich zum Beispiel der Winkel unter dem sich zwei Geraden schneiden bestimmen.

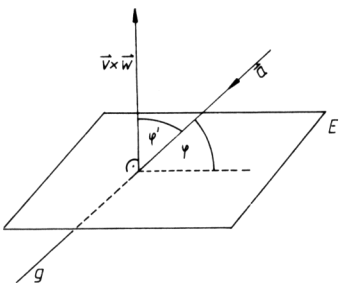
Sei g1: $\vec{X} = \vec{A} + k\vec{u}$
g2: $\vec{X} = \vec{B} + l\vec{v}$

Dann gilt für den Schnittwinkel δ der Geraden g1 und g2:

$$\cos\delta = \frac{\vec{u} \circ \vec{v}}{|\vec{u}|\,|\vec{v}|}$$

(2) Winkel zwischen Ebene und Gerade

Gegeben sei die Ebene E: $\vec{X} = \vec{A} + k\vec{u} + l\vec{v}$, sowie eine Gerade g: $\vec{X} = \vec{B} + r\vec{w}$



Durch $\overline{u}\times\overline{v}$ erhält man einen Vektor der senkrecht auf der von $\overline{u}$ und $\overline{v}$ aufgespannten Ebene steht. Für den Winkel φ zwischen der Geraden g und der Ebene E gilt dann folgende Beziehung:

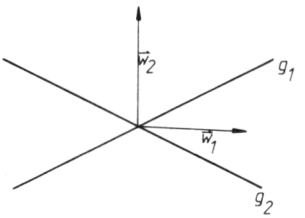
$$\varphi = 90° - \varphi'$$

der Winkel φ' läßt sich aber ganz leicht anhand des Skalarprodukt von $\overline{u}\times\overline{w}$ und $\overline{w}$ ermitteln,da gilt:

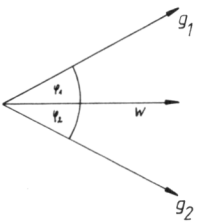
$$\cos\varphi' = \frac{(\overline{u}\times\overline{v})\overline{w}}{|\overline{u}\times\overline{v}|\,|\overline{w}|}$$

(3) Winkelhalbierende zweier Geraden

Gegeben sind die Richtungsvektoren zweier Geraden. Gesucht werden die Richtungsvektoren der beiden Winkelhalbierenden $\overline{w1}$ und $\overline{w2}$ dieser Geraden.



Wir gehen aus vom Skalarprodukt:



Da $\overline{w}$ Winkelhalbierende von g1 und g2 ist, muß gelten $\phi1 = \phi2 \iff \cos\phi1 = \cos\phi2$. Anstelle von $\cos\phi1 = \cos\phi2$ können wir mir Hilfe des Skalarprodukts schreiben:

$$\frac{\vec{w} \circ \vec{a}}{|\vec{w}|\,|\vec{a}|} = \frac{\vec{w} \circ \vec{b}}{|\vec{w}|\,|\vec{b}|}$$

$$\frac{\vec{w} \circ \vec{a}}{|\vec{a}|} = \frac{\vec{w} \circ \vec{b}}{|\vec{b}|}$$

wobei $\vec{a}$,$\vec{b}$ die Richtungsvektoren von g1 und g2 sind.
Da $\vec{w}$ in einer Ebene mit $\vec{a}$ und $\vec{b}$ liegt, läßt sich $\vec{w}$ als Linearkombination von $\vec{a}$ und $\vec{b}$ darstellen. Es muss gelten:

$$\vec{w} = k\vec{a} +/- \vec{b} \quad (*)$$

Dann gilt:

$$\frac{(k\vec{a} +/- \vec{b}) \circ \vec{a}}{|\vec{a}|} = \frac{(k\vec{a} +/- \vec{b}) \circ \vec{b}}{|\vec{b}|}$$

$$<=> \quad \frac{k\vec{a} +/- \vec{ab}}{|\vec{a}|} = \frac{k\vec{ab} +/- \vec{b}}{|\vec{b}|}$$

$$<=> \quad \frac{k\vec{a}}{|\vec{a}|} +/- \frac{\vec{ab}}{|\vec{a}|} = \frac{k\vec{ab}}{|\vec{b}|} +/- \frac{\vec{b}}{|\vec{b}|}$$

$$<=> \quad k|\vec{a}| +/- \frac{\vec{ab}}{|\vec{a}|} = \frac{k\vec{ab}}{|\vec{b}|} +/- |\vec{b}|$$

$$<=> \quad k\left(|\vec{a}| +/- \frac{\vec{ab}}{|\vec{b}|}\right) = |\vec{b}| +/- \frac{\vec{ab}}{|\vec{a}|}$$

$$<=> k\left(\frac{|\vec{a}|\,|\vec{b}| +/- \vec{ab}}{|\vec{b}|}\right) = \left(\frac{|\vec{b}|\,|\vec{a}| +/- \vec{ab}}{|\vec{a}|}\right)$$

$$<=> \quad k = \frac{|\vec{b}|}{|\vec{a}|}$$

Einsetzen in Gleichung (*) liefert dann die Lösung:

$$\vec{w}1/2 = \frac{|\vec{b}|}{|\vec{a}|}\vec{a} +/- \vec{b} = \frac{\vec{a}}{|\vec{a}|} +/- \frac{\vec{b}}{|\vec{b}|} = \vec{a0} +/- \vec{b0}$$

Die Winkelhalbierenden erhält man also durch Addition bzw. Subtraktion der zu $\vec{a}$ und $\vec{b}$

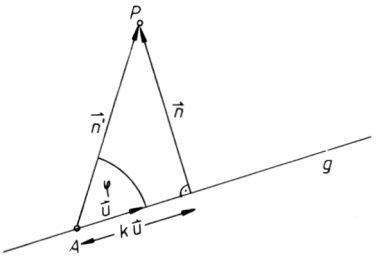
gehörigen Einheitsvektoren.

(4) Abstand eines Punktes von einer Geraden

Zu einer Geraden orthogonale (=senkrechte) Vektoren kann man dazu benutzen um den Abstand eines Punktes von einer Geraden zu bestimmen.
Es seien ein Punkt P sowie eine Gerade g (Trägerpunkt A, Richtungsvektor $\overline{u}$) gegeben.

$$g: \overline{X} = \overline{A}+k\overline{u}$$

Betrachten wir hierzu folgende Skizze:



Den kürzesten Abstand eines Punktes von einer Geraden erhalten wir wenn wir die Länge des Lotes von P auf g bestimmen.
Es gilt offensichtlich:

$$\overline{n}' = k\overline{u}+\overline{n}$$

Den Wert für $k\overline{u}$ können wir ganz einfach berechnen indem wir die Projektion von $\overline{n}'$ in $\overline{u}$ bestimmen.Es gilt:

$$\frac{\overline{n}'\circ\overline{u}}{|\overline{u}|}\overline{u} = k\overline{u}$$

Aus obiger Skizze folgt:

$$|k\overline{u}| = l$$

$$l = |\overline{n}'|\cos\varphi$$

$$l = |\overline{n}'|\frac{|\overline{u}|}{|\overline{u}|}\cos\varphi = \frac{\overline{n}'\circ\overline{u}}{|\overline{u}|}$$

$$K\vec{u} = l\vec{u}\vec{0} = \frac{K}{|\vec{u}|}\vec{u}$$

$$K\vec{u} = \frac{\vec{n}'\circ\vec{u}}{|\vec{u}|}\,\frac{\vec{u}}{|\vec{u}|} = \frac{\vec{n}'\circ\vec{u}}{|\vec{u}|}\,\vec{u} = \frac{\vec{n}'\circ\vec{u}}{\vec{u}^2}\,\vec{u}$$

Einsetzen ergibt dann:

$$\vec{n}' = \vec{n} + \frac{\vec{n}'\circ\vec{u}}{\vec{u}^2}\,\vec{u}$$

$$\Leftrightarrow \vec{n} = \vec{n}' - \frac{\vec{n}'\circ\vec{u}}{\vec{u}^2}\,\vec{u}$$

Der Abstand eines Punktes von einer Geraden läßt sich also folgendermaßen berechnen:

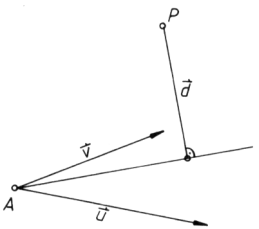
$$d = |\vec{m}| = \left| \vec{n}' - \frac{\vec{n}'\circ\vec{u}}{\vec{u}^2}\,\vec{u} \right|$$

(5) Abstand eines Punktes von einer Ebene

Normalenform der Ebenendarstellung

Schnitt Gerade - Ebene

Gegeben sei ein Punkt P aus E sowie eine Ebene E: $\vec{X} = \vec{A} + K\vec{u} + l\vec{v}$



Dieses Problem ist nun schon etwas schwieriger geartet als das vorhergehende. Es läßt sich besonders einfach lösen, wenn wir zu einer anderen Schreibweise der Ebene übergehen:

Gegeben sei E: $\overline{x} = \begin{bmatrix} a1 \\ a2 \\ a3 \end{bmatrix} + K \begin{bmatrix} u1 \\ u2 \\ u3 \end{bmatrix} + l \begin{bmatrix} v1 \\ v2 \\ v3 \end{bmatrix}$

Dies führt zu den 3 Gleichungen

(1) x1 = a1+Ku1+lv1
(2) x2 = a2+Ku2+lv2
(3) x3 = a3+Ku3+lu3

Die Lösung dieses Gleichungssystems führt auf die sog. parameterfreie Form der Ebenengleichung:

(1) x1-a1 = Ku1+lv1
(2) x2-a2 = Ku2+lv2
(3) x3-a3 = Ku3+lv3

Wir setzen für das folgende voraus, daß u1 ungleich 0 ist.

$$(2)+(-\frac{u2}{u1})(1)=(1)':$$

$$(x2-a2)-(x1-a1)(\frac{u2}{u1}) = l(v2-v1\frac{u2}{u1}) = l(\frac{v2u1-v1u2}{u1})$$

$$(3)+(-\frac{u3}{u1})(1)=(2)':$$

$$(x3-a3)-(x1-a1)(\frac{u3}{u1}) = l(v3-v1\frac{u3}{u1}) = l(\frac{v3u1-v1u3}{u1})$$

$$(2)'-(\frac{v3u1-v1u3}{v2u1-v1u2})(1)':$$

$$(x3-a3)-(x1-a1)(\frac{u3}{u1})-(x2-a2)(\frac{v3u1-v1u3}{v2u1-v1u2})$$

$$+(x1-a1)(\frac{u2}{u1})(\frac{v3u1-v1u3}{v2u1-v1u2}) = 0$$

Umordnen ergibt:

$$x3-x1\frac{u3}{u1} - x2(\frac{v3u1-v1u3}{v2u1-v1u2})+x1(\frac{u2}{u1})(\frac{v3u1-v1u3}{v2u1+v1u2})-C = 0$$

Die Konstante C läßt sich später, wie wir sehen werden sehr einfach ermitteln. Wir bringen nun noch alle Terme auf einen Nenner:

$$x3u1(v2u1-v1u2)-x1u3(v2u1-v1u2)-$$
$$x2u1(v3u1-v1u3)+x1u2(v3u1-v1u3)+C' = 0$$

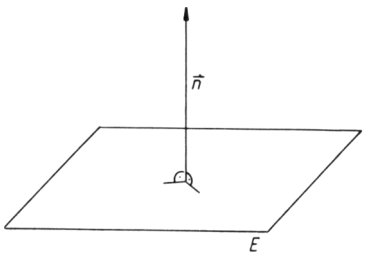
$$x3u1(v2u1-v1u2)-x2u3(v3u1-v1u3)+$$
$$x1(v3u1u2-v1u3u2-v2u3u1+v1u2u3)+C' = 0$$

$$x3u1(v2u1-v1u2)-x2u1(v3u1-v1u3)+$$
$$x1u1(v3u2+v2u3)+C'=0$$

Division durch u1 liefert:

$$x1(v3u2-v2u3)+x2(v1u3-v3u1)+$$
$$x3(v2u1-v1u2)+C'' = 0 \quad (*)$$

Dieses Ergebnis war zu erwarten.Warum werden sie sich fragen.



Betrachten wir die Ebene E. Offensichtlich gibt es nur EINE Klasse von Vektoren, die auf dieser Ebene senkrecht stehen. Kennt man einen Vektor, der senkrecht auf E steht, so kennt man alle, da alle anderen von diesem einen Vektor linear abhängig sind.
Die Lage der Ebene im Raum wird also allein durch einen Trägerpunkt A und einen Vektor, der senkrecht zu ihr steht, beschrieben. Den senkrechten Vektor bezeichnet man als Normalenvektor der Ebene.
Da es egal ist, was für einen Normalenvektor man verwendet, ist z.B. der Vektor uxv ein Normalenvektor von E.

$$\overline{u}\times\overline{v} = \begin{bmatrix} u1 \\ u2 \\ u3 \end{bmatrix} \times \begin{bmatrix} v1 \\ v2 \\ v3 \end{bmatrix} = \begin{bmatrix} u2v3-u3v2 \\ u3v1-u1v3 \\ u1v2-u2v1 \end{bmatrix} = \begin{bmatrix} n1 \\ n2 \\ n3 \end{bmatrix} = \overline{n}$$

Durch Vergleich der Koeffizienten von Gleichung (*) mit den Komponenten von $\overline{n}$ stellen wir fest:

$$E: n1x2+n2x2+n3x3+n0 = 0 \quad (**) \quad ,n0 = \text{const.}$$

Die Konstante n0 läßt sich ganz leicht ermitteln, indem wir den Trägerpunkt A in die Gleichung einsetzen.
Es gilt:

$$n1a1+n2a2+n3a3+n0 = 0$$

$$\Rightarrow n0 = -(n1a1+n2a2+n3a3)$$

Man bezeichnet die Form (**) als Normalenform der Ebenengleichung.
Was nützt uns dies nun für die Abstandsbestimmung des Punktes P von der Ebene? Sehr viel, denn wenn wir die parameterfreie Form der Ebenendarstellung kennen, ist es besonders einfach den Schnittpunkt einer Geraden mit dieser Ebene zu bestimmen.
Die Lotgerade von P auf E lautet:

$$g: \overline{X} = \overline{P}+r\overline{n}$$

Gleichsetzen von g und E liefert den Schnittpunkt S.

$$\overline{P}+r\overline{n} = \overline{A}+k\overline{u}+l\overline{v}$$

Das Ergebnis haben wir bereits bestimmt. Wir brauchen nur in Gleichung (**) anstelle von $\overline{X}$ den Punkt $\overline{P}+r\overline{n}$ der Geraden g einsetzen.
Dies liefert:

$$n1(p1+rn1)+n2(p2+rn2)+n3(p3+rn3)+n0 = 0$$
$$n1p1+rn1^2+n2p2+rn2^2+n3p3+rn3^2+n0 = 0$$
$$n1p1+n2p2+n3p3+n0 = r(n1^2+n2^2+n3^2)$$

$$\Leftrightarrow r = \frac{n1p2+n2p2+n3p3+n0}{n1^2+n2^2+n3^2}$$

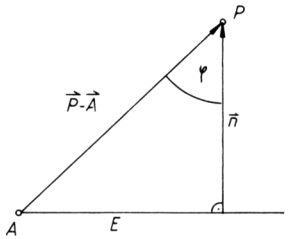
Für die Länge des Lotes l gilt aber folgende Beziehung:

$$l = |r|\overline{n}|$$

da $|\overline{n}|^2 = n1^2+n2^2+n3^2$ ist gilt

$$l = |\frac{n1p1+n2p2+n3p3+n0}{\sqrt{n1^2+n2^2+n3^2}}| \quad (***)$$

Dies stimmt aber mit dem Ergebnis der Vektorrechnung überein, das wir erhalten, wenn wir den Verbindungsvektor $\overline{P}-\overline{A}$ in den Normalenvektor projezieren, was ja identisch mit der Bestimmung der Länge des Lotes ist.



$$\frac{(\overline{P}-\overline{A})\circ\overline{n}}{|\overline{n}|} = 1$$

Die Gleichung (***) für $\overline{X}$ geschrieben ist nichts anderes als eine transformierte Form der Normalenform der Ebenegleichung. Sie wird zu Ehren des deutschen Mathematikers O.L. Hesse (1811-1874) Hessenormalenform der Ebenegleichung genannt.
Die Hessenormalenform wird oft auch vektoriell geschrieben:

$$\overline{n0}\circ(\overline{X}-\overline{A}) = 0$$

Man kann der Hesse'schen Normalenform, wenn man das Vorzeichen von n0 negativ hält, auch entnehmen,auf welcher Seite der Ebene der Punkt liegt. Durch das negative Vorzeichen wird n0 immer so orientiert, daß es vom Ursprung O des Systems wegweist.
Setzt man dann in

$$\frac{n1\times 1+n2\times 2+n3\times 3+n0}{(-\mathrm{sgn}\ n0)\sqrt{n1^2+n2^2+n3^2}} = d$$

ein,so erhält man, je nach Lage von P unterschiedliche Vorzeichen von d.Es gilt:
d>0 : Ursprung und P liegen auf verschiedenen Seiten von E
d=0 : P E
d<0 : Ursprung und P liegen auf der selben Seite von E

Der Vektorraum

Wir sind bisher mehr oder weniger formlos im mathematischen Sinne vorgegangen und haben einige sehr wichtige Begriffe mehr oder weniger übergangen. Dennoch dürften sie jetzt eine ungefähre Vorstellung von dem haben, was (anschaulich) ein Vektorraum ist. Ein Teil des nun folgenden Kapitels wird nun sehr formal sein. Leider kann aber nicht darauf verzichtet werden. Wir werden jedoch nach Möglichkeit immer eine anschauliche Deutung zu dem Gesagten geben.

Definition: Eine nichtleere Menge V, für deren Elemente eine Addition (+) und eine Multiplikation (*) mit reellen Zahlen erklärt ist, heißt reeller Vektoraum V=(V,+,*). Seine Elemente heißen Vektoren. (Anschaulich ist ein Vektorraum V das,was wir bisher mit R bezeichnet haben).
Es müßen die folgenden Axiome erfüllt sein:

(+)
V1: $\vec{a}+\vec{b} = \vec{c}$ $\quad \vec{a},\vec{b} \in V \Rightarrow \vec{a}+\vec{b}=\vec{c} \in V$
Abgeschlossenheit bzgl. der Addition

(Wählen wir zwei Vektoren $\vec{a}$ und $\vec{b}$ aus dem R^n, so muß auch die Summe aus $\vec{a}$ und $\vec{b}$ im R^n liegen. Von dieser Tatsache sind wir bisher stillschweigend ausgegangen.)

V2: $(\vec{a}+\vec{b})+\vec{c} = \vec{a}+(\vec{b}+\vec{c})$
Assoziativgesetz

(Diese Voraussetzung braucht i.a. nicht nachgeprüft zu werden, da sie sich ganz allgemein für jede Verknüpfung, wie immer man sie definiert, beweisen läßt.)

V3: $\vec{a}+\vec{b} = \vec{b}+\vec{a}$ $\quad \vec{a},\vec{b} \in V$
Kommutativität

(Diese Voraussetzung war bisher immer erfüllt wie wir bereits gesehen haben. Sie ist aber keineswegs selbstverständlich, sondern i.a. die Ausnahme von der Regel (vgl. Kreuzprodukt))

V4: $\vec{a}+\vec{o} = \vec{a}$
Nullelement

(Die Existenz eines Nullelements ist sehr wichtig

wie wir bei V5 gleich sehen werden. Es verhält sich bei der Addition neutral.)

V5: $\vec{a}+\vec{a} = \vec{0}$
Inverses Element

(Es gibt für alle $\vec{a}$ V ein $\vec{a}$ (Inverses Element zu $\vec{a}$), für das V5 gilt. Wählen wir z.B. den Vektor $\vec{a} \in R^n$, so ist sein Inverses gegeben durch $-\vec{a}$, denn $\vec{a}-\vec{a} = \vec{o}$ erfüllt V5. Dies kann aber nicht erfüllt werden, wenn kein Nullelement vorhanden ist.
Die Folge davon wäre, daß alle Rechengesetze ungültig wären, da dann z.B. gelten könnte:

$\vec{a}-\vec{a} \neq \vec{b}-\vec{b}$)

(*)

V6: $k\vec{a} = \vec{b}$,$k \in R$
Abgeschlossenheit bzgl. der Multiplikation mit einer reellen Zahl

(Wählen wir eine Vektor $\vec{a} \in V$, so muß auch ein beliebiges Vielfaches dieses Vektors $\in V$ sein.)

V7: $1\vec{a} = \vec{a}$
Einselement

(Die Multiplikation mit der Eins darf den Vektor nicht verändern. Für uns ist das unerheblich, da wir nur mit reellen Zahlen und den Standardvektorräumen arbeiten. Der Vollständigkeit halber sei aber erwähnt, daß man die Multiplikation so wählen kann, daß das Einselement den Vektor ändert.)

V8: $k(\vec{a}+\vec{b}) = k\vec{a}+k\vec{b}$
$(k+l)\vec{a} = k\vec{a}+l\vec{a}$
Distributivität

Dies alles mag den Eindruck erwecken, daß es sich hierbei tatsächlich um einen 'Wust' von Formalitäten handelt. Tatsache aber ist, daß es sich dabei um absolute Mindestanforderungen handelt. Ist eines dieser Axiome nicht mehr erfüllt, so ist es nicht mehr möglich, wie in den vorangegangenen Abschnitten demonstriert mit Vektoren zu rechnen, denn es könnte, um nur ein Beispiel zu geben, gelten: $\vec{a}+\vec{a} \neq 2\vec{a}$, womit alle bisherigen Rechnungen hinfällig wären.
Sie können diese Axiome auch gleich wieder getrost vergessen, da sie in den von uns

behandelten Vektorräumen immer erfüllt sind. Der Vollständigkeit halber erschien es mir aber angebracht, diesen Abschnitt hier einzuschieben, da es einen wagen Eindruck davon vermittelt, wieviel Mathematik hinter dem, was wir hier so einfach hinnehmen,steht.

Basis und Dimension

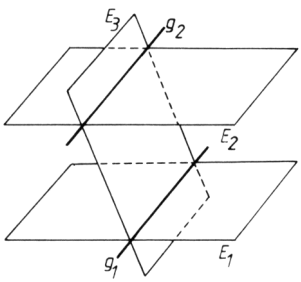
Als Basis bezeichnet man eine unverkürzbare Menge von linear unabhängigen Vektoren aus V, aus denen sich alle Elemente von V durch Linearkombinationen der Basisvektoren darstellen lassen. Die Anzahl der linear unabhängigen Basisvektoren bezeichnet man als Dimension dimV von V.

Das Gauß'sche Eliminationsverfahren

Für das Rechnen mit Matrizen ist das Gauß'sche Eliminationsverfahren von enormer Wichtigkeit. Es ist aber auch von anderen Standpunkten aus gesehen sehr intersessant. So ist es z.B. das Standardverfahren zur Lösung von linearen Gleichungssystemen.
Betrachten wir hierzu das folgende Problem:
Gegeben seien 3 Ebenen E1,E2,E3 des R^3.Wir wollen nun eine Aussage darüber treffen, wie diese Ebenen zueinander liegen. Wir können folgende Fälle unterscheiden:

a) Die 3 Ebenen liegen parallel.

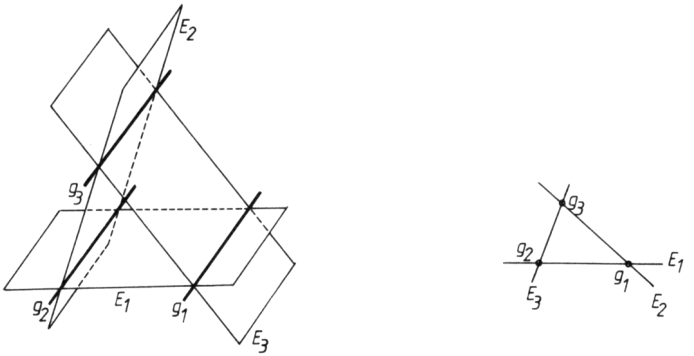
b) 2 Ebenen liegen parallel.Dann schneidet die dritte Ebene die beiden anderen Ebenen in 2 Schnittgeraden g1 und g2.



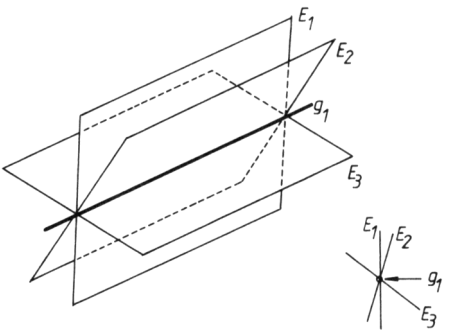
c) Die 3 Ebenen liegen so, daß keine zu einer

anderen parallel ist. Dann müßen wir wieder zwischen 3 Fällen unterscheiden.

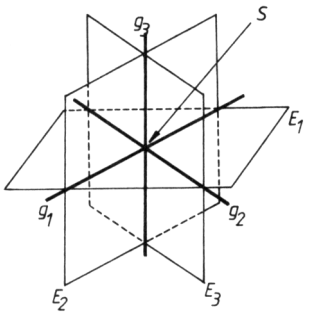
c1) Die 3 Ebenen liegen so, daß sich je 2 in 3 Schnittgeraden g1, g2 und g3 schneiden.



c2) Die 3 Ebenen schneiden sich in einer Geraden g1.



c3) Die 3 Ebenen schneiden sich so, daß sich die Schnittgeraden g1, g2, g3 in einem Punkt schneiden.



Wie Können wir nun darüber Aussagen treffen?
Wie wir an früherer Stelle gesehen haben, läßt sich die Lage einer Ebene durch ihren Normalenvektor charakterisieren.
Die 3 Ebenen lassen sich in der Form

E1: a11x1+a12x2+a13x3+a14 = 0
E2: a21x1+a22x2+a23x3+a24 = 0
E3: a31x1+a32x2+a33x3+a34 = 0

schreiben,wobei die Vektoren

$$\begin{bmatrix} a11 \\ a12 \\ a13 \end{bmatrix}, \begin{bmatrix} a21 \\ a22 \\ a23 \end{bmatrix}, \begin{bmatrix} a31 \\ a32 \\ a33 \end{bmatrix}$$

die Normalenvektoren der 3 Ebenen sind.
Wir Können jetzt Informationen über die Lage der Ebenen erhalten, wenn wir diese Normalenvektoren auf lineare Abhängigkeit untersuchen, denn wenn alle 3 Ebenen z.B. parallel liegen, so müßen sich alle Normalenvektoren durch einen einzigen von den Dreien darstellen lassen.
Die Bestimmung der Lage der Ebenen zueinander läuft letztendlich auf die Lösung des linearen Gleichungssystems hinaus. Wie man ein derartiges System löst, werden wir jetzt sehen.
Wir schreiben hierzu die Koeffizienten in ein rechteckiges Schema.

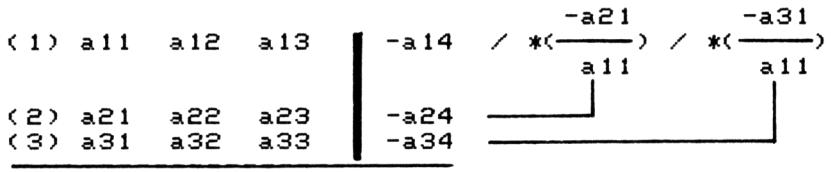
$$\left[\begin{array}{ccc|c} a11 & a12 & a13 & -a14 \\ a21 & a22 & a23 & -a24 \\ a31 & a32 & a33 & -a34 \end{array}\right]$$

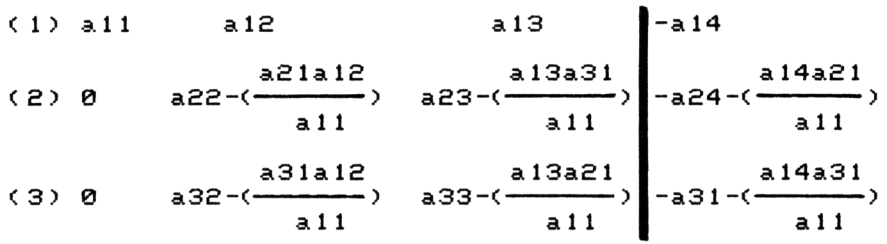
Ein derartiges Schema bezeichnet man als Matrix.
Weiter steht überhaupt nichts hinter diesem

Begriff.

Zur Lösung des Systems benötigen wir eine Zeile mit möglichst vielen Nullen. Dazu bringen wir das System in die sog. Zeilen-Stufen-Form:

a11 # 0





Wir haben hierzu die erste Zeile mit einem reellen Wert (-a21/a11) bzw. (-a31/a11) multipliziert und zur Zeile (2) bzw. (3) addiert. Wie sich mathematisch leicht zeigen läßt wird die Lösungsmenge des Systems dadurch nicht verändert. Dieses Verfahren bezeichnet man als elementare Zeilenumformungen. Man unterscheidet:

(1) Vertauschen zweier Zeilen
(2) Multiplikation einer Zeile mit einem Skalar #0
(3) Addition zweier Zeilen

Wir haben damit in der ersten Spalte der Matrix in den Zeilen (2) und (3) Nullen erzeugt. Nach dem selben Verfahren erzeugen wir nun durch Multiplikation von Zeile (2) mit einem geeigneten Skalar eine Null an der zweiten Stelle der dritten Zeile. Wir erhalten dann folgendes Schema:

```
c11 c12 c13 | c14
 0   c22 c23 | c24
 0    0  c33 | c34
```

Von unten beginnend löst man dann das Gleichungssystem sukzessive nach oben auf.(c33x3=c34 ⇒ x3=c34/c32 → in (2) usw.). Am

besten verständlich wird es wohl an einem Beispiel:

Gegeben seien die 3 Ebenen E1,E2,E3 mit

$$\begin{aligned} E1:&\ 2\,x1 + x2 + 3\,x3 - 9 = 0\\ E2:&\ x1 - 2\,x2 + x3 + 2 = 0\\ E3:&\ 3\,x1 + 2\,x2 + 2\,x3 - 7 = 0 \end{aligned}$$

Wir bestimmen die Lösung nach dem Gauß'schen Eliminationsverfahren.

$$\begin{array}{rrrrl} 2 & 1 & 3 & 9 & \\ 1 & -2 & 1 & -2 & \text{Vertauschen} \\ 3 & 2 & 2 & 7 & \\ \hline 1 & -2 & 1 & -2 & /(-2)\ /(-3) \\ 2 & 1 & 3 & 9 & \\ 3 & 2 & 2 & 7 & \\ \hline 1 & -2 & 1 & -2 & \\ 0 & 5 & 1 & 13 & /(-\frac{8}{5}) \\ 0 & 8 & -1 & 13 & \\ \hline (1)\ 1 & -2 & 1 & -2 & \\ (2)\ 0 & 5 & 1 & 13 & \\ (3)\ 0 & 0 & -(\frac{13}{5}) & -(\frac{39}{5}) & \\ \hline \end{array}$$

$$-\frac{13}{5}x3 = -\frac{39}{5} \qquad \Rightarrow x3 = 3$$

in (2) : $5x2 + 3 = 13 \quad \Rightarrow x2 = 2$
in (1) : $x1 - 4 + 3 = -2 \quad \Rightarrow x1 = -1$

Wir erhalten also als Lösung den Punkt

$$\begin{bmatrix} -1 \\ 2 \\ 3 \end{bmatrix}$$

der auf allen 3 Ebenen liegt.
Wie würde aber das Ergebnis aussehen, wenn wir E3 wie folgt wählen würden:

$$E3:\ 3\,x1 - x2 + 4\,x3 - 7 = 0$$

Nach dem Gauß'schen Eliminationsverfahren kommt man hier auf

$$\begin{array}{cccc} 1 & -2 & 1 & -2 \\ 0 & 5 & 1 & 13 \\ 0 & 0 & 0 & 13 \end{array}$$

was offensichtlich zu dem Widerspruch 0 = 13 führt.
Betrachten wir uns doch die Normalenvektoren der Ebene einmal genauer:

$$\vec{n}1 = \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}, \vec{n}2 = \begin{bmatrix} 2 \\ 1 \\ 3 \end{bmatrix}, \vec{n}3 = \begin{bmatrix} 3 \\ -1 \\ 4 \end{bmatrix}$$

Offensichtlich gilt : $\vec{n}1 + \vec{n}2 = \vec{n}3$
d.h. die Normalenvektoren sind linear abhängig. Der Widerspruch 0 = 13 führt aber dazu, daß wir sagen können: Es gibt keinen Punkt, den alle 3 Ebenen gemeinsam haben. Daraus können wir aber nach Fall c1) sagen, daß die Schnittgeraden je zweier Ebenen parallel verlaufen.
Würde hingegen in Zeile (3) 0 = 0 entstehen, so ließe sich das System noch in Abhängigkeit von einem Parameter lösen. Es gälte:

$$\begin{aligned} x1 - 2\,x2 + x3 &= -2 \\ 5\,x2 + x3 &= 13 \end{aligned}$$

Wir setzen für x3 den Parameter K ein und erhalten:

$$5x2 + K = 13 \quad <=> \quad x2 = \frac{13-K}{5}$$

$$x1 - 2\left(\frac{13-K}{5}\right) + K = -2 \quad <=> \quad x1 - \frac{26}{5} + \frac{2K}{5} + K = -2$$

$$<=> x1 = \frac{16}{5} - \frac{7}{5}K$$

Vektoriell geschrieben sieht das so aus:

$$\begin{bmatrix} x1 \\ x2 \\ x3 \end{bmatrix} = \begin{bmatrix} 16/5 \\ 13/5 \\ 0 \end{bmatrix} + K \begin{bmatrix} -7/5 \\ -1/5 \\ 1 \end{bmatrix}$$

Wir ziehen noch den Wert 1/5 heraus und erhalten

$$\begin{bmatrix} x1 \\ x2 \\ x3 \end{bmatrix} = \begin{bmatrix} 16/5 \\ 13/5 \\ 0 \end{bmatrix} + K \begin{bmatrix} -7 \\ -1 \\ 5 \end{bmatrix}$$

Dies ist die Gleichung einer Geraden. Die 3

Ebenen schneiden sich also in einer Geraden (Fall c2))

Dieses Schema läßt sich nun verallgemeinern:
Schreibt man die Ebenen in Matrizenform, wie oben, so läßt sich über elementare SPALTENumformungen der sog. Rang RgA einer Matrix A bestimmen. (Anm.: Es ist egal ob man Spalten- oder Zeilenumformungen macht, denn es gilt Zeilenrang = Spaltenrang. Man darf allerdings nur jeweils eine Form, ENTWEDER Zeilen- ODER Spaltenumformungen vornehmen. Im Hinblick auf die Lösung des Gleichungssystems sind allerdings elementare Spaltenumformungen vorzuziehen.)
Man definiert hierzu den Rang RgA einer Matrix A wie folgt:
Unter dem Rang einer Matrix A, in Zeichen RgA, versteht man die Maximalzahl der linear unabhängigen Spaltenvektoren. Man bezeichnet den Rang der Koeffizientenmatrix mit RgA, den der erweiterten mit $Rg\tilde{A}$.

$$A = \begin{bmatrix} a11 & a12 & a13 \\ a21 & a22 & a23 \\ a31 & a32 & a33 \end{bmatrix}$$

$$\tilde{A} = \begin{bmatrix} a11 & a12 & a13 & a14 \\ a21 & a22 & a23 & a24 \\ a31 & a32 & a33 & a34 \end{bmatrix}$$

Allgemein gilt dann folgendes Schema:

	RgA	$Rg\tilde{A}$	Lösung
(1)	3	3	Punkt
(2)	2	3	parallele Geraden (2 oder 3)
(3)	2	2	1 gemeinsame Gerade
(4)	1	2	3 parallele Ebenen,kein Schnitt
(5)	1	1	die 3 Ebenen sind identisch
(6)	0	1	Keine Ebene !

Das bisher gesagte sollte in erster Linie dazu dienen, verständlich zu machen, was man unter einer Matrix zu verstehen hat. Eine Matrix ist, wie ein Vektor, natürlich nicht nur eine andere Notation von etwas bekanntem. Ihre wahre Bedeutung wird erst dadurch deutlich, daß man verschiedene algebraische Operationen mit ihnen durchführen kann. Auf diese wollen wir nun näher eingehen.

Verknüpfungen von Matrizen

Man bezeichnet die Menge aller Matrizen mit m

Zeilen und n Spalten als Mat(mxn,R).
Sei A eine mxn Matrix.

$$A = \begin{bmatrix} a11 & a12 & \dots & a1n \\ a21 & a22 & \dots & a2n \\ & \vdots & & \\ am1 & am2 & \dots & amn \end{bmatrix}$$

Dann seien die Verknüpfungen (+) und (*) wie folgt erklärt:
A aus Mat(mxn,R) ;B aus Mat(nxp,R)

$$A = \begin{bmatrix} a11 & a12 & \dots & a1n \\ a21 & a22 & \dots & a2n \\ & \vdots & & \\ am1 & am2 & \dots & amn \end{bmatrix}, B = \begin{bmatrix} b11 & b12 & \dots & b1p \\ b21 & b22 & \dots & b2p \\ & \vdots & & \\ bn1 & bn2 & \dots & bnp \end{bmatrix}$$

Dann sei die Produktmatrix C wie folgt erklärt:

$$C = A*B = \begin{bmatrix} c11 & \dots & c1p \\ \vdots & & \vdots \\ cm1 & \dots & cmp \end{bmatrix} \quad \text{aus Mat(mxp,R)}$$

mit

$$cij = \begin{bmatrix} ai1 \\ ai2 \\ \vdots \\ ain \end{bmatrix} o \begin{bmatrix} b1j \\ b2j \\ \vdots \\ bnj \end{bmatrix}$$

Anschaulich gesprochen: Man erhält das Element der Produktmatrix, das an der Schnittstelle von i-ter Zeile und j-ter Spalte steht durch Skalarmultiplikation der i-ten Zeile von A mit der j-ten Spalte von B.
Es ist daher unverzichtbar, daß die Spaltenanzahl von A und die Zeilenanzahl von B übereinstimmen.

Seien A,B Matrizen aus Mat(mxn,R)
Dann ist A+B = B+A =

$$\begin{bmatrix} a11+b11 & a12+b12 & \dots & a1n+b1n \\ & \vdots & & \\ am1+bm1 & am2+bm2 & \dots & amn+bmn \end{bmatrix}$$

Nullelement:

$$\begin{bmatrix} 0 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 0 \end{bmatrix}$$

Neutrales Element:

$$\begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & 1 \end{bmatrix}$$

Multiplikation mit einem Skalar K:

$$KA = \begin{bmatrix} Ka11 & Ka12 & \dots & Ka1n \\ \vdots & & & \vdots \\ Kam1 & Kam2 & \dots & Kamn \end{bmatrix}$$

Es gelten die folgenden Gesetze:

M1: $(K+1)A = KA+1A$
$K(A+B) = KA+1B$

M2: $(AB)C = A(BC)$

M3: $AB \# BA$ (im allgemeinen !)

M4: Es gibt Nullteiler, d.h. Matrizen $A\#0$, $B\#0$ mit $AB=0$

z.B. $\begin{bmatrix} 5 & 2 & -2 & 3 \\ 9 & 2 & -3 & 4 \end{bmatrix} * \begin{bmatrix} 2 & 2 & 2 \\ -1 & 3 & -5 \\ 16 & 8 & 24 \\ 8 & 0 & 16 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

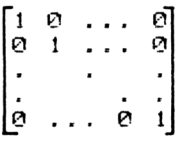
Folglich gilt:

(nicht !)

$AB = 0 \quad ,A \# 0 \quad \Rightarrow \quad B = 0$ (i.a.)
$AB = AC \ ,A \# 0 \quad \Rightarrow \quad B = C$ (i.a.)

Eine Matrix der Form

$$\begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & 1 \end{bmatrix}$$



bezeichnet man als n-reihige quadratische Einheitsmatrix En.
Matrizen der Form Mat(nxn,R) bezeichnet man als quadratische Matrizen Mat(n,R).

In der Menge der Matrizen gibt es eine weitere Form der Rechenoperation, die in dieser Form bisher nirgends aufgetaucht ist, die sog. Transponierte Matrix ${}^tA$. Sie entsteht aus der Matrix A durch Umwandlung der Zeilenvektoren in Spaltenvektoren.

$$ {}^t\begin{bmatrix} 2 & -1 & 3 & 6 \\ 1 & 0 & 5 & 3 \\ 3 & 5 & -4 & 2 \end{bmatrix} = \begin{bmatrix} 2 & -1 & 3 \\ -1 & 0 & 5 \\ 3 & 5 & -4 \\ 6 & 3 & 2 \end{bmatrix} $$

Eine sehr wichtige Eigenschaft von quadratischen Matrizen ist ihre Invertierbarkeit. Voraussetzung dazu ist allerdings, daß gilt:
Ist A aus Mat(n,R), dann muß RgA = n sein. Warum, werden wir noch sehen.
Als Inverse Matrix A^{-1} einer Matrix A bezeichnet man diejenige Matrix für die folgendes gilt:

$$A*A^{-1} = En$$
$$A^{-1}*A = En$$

Wichtig ist hierbei die Kommutativität, die ja für die Matrizenmultiplikation allgemein nicht gilt.
Bevor wir auf die Berechnung der Inversen Matrix eingehen, noch ein paar Worte zur Matrizenmultiplikation.

Lineare Abbildungen

Als lineare Abbildungen bezeichnet man Abbildungen

$$F:V->W$$

mit

$$L1: F(\vec{v}+\vec{w}) = F(\vec{v})+F(\vec{w})$$
$$L2: F(k\vec{v}) = kF(\vec{v})$$

Eine besondere Art der Matrizenmultiplikation ist folgende:

$$\begin{bmatrix} a11 & \dots & a1n \\ . & & . \\ . & & . \\ . & & . \\ am1 & \dots & amn \end{bmatrix} * \begin{bmatrix} x1 \\ x2 \\ . \\ . \\ xn \end{bmatrix}$$

Dabei besteht die zweite Matrix lediglich aus einer Spalte, sie ist also ein Vektor. Multipliziert man aus, so sieht man sofort, daß auch das Ergebnis wieder ein Vektor ist.

$$\begin{bmatrix} a11 & \dots & a1n \\ . & & . \\ . & & . \\ . & & . \\ am1 & \dots & amn \end{bmatrix} * \begin{bmatrix} x1 \\ x2 \\ . \\ . \\ xn \end{bmatrix} = \begin{bmatrix} a11x1+a12x2+\dots+a1nxn \\ a21x1+a22x2+\dots+a2nxn \\ . \\ . \\ am1x1+am2x2+\dots+amnx2 \end{bmatrix} = \begin{bmatrix} y1 \\ y2 \\ . \\ . \\ yn \end{bmatrix}$$

In Matrizenschreibweise schreibt sich das dann so:

$$AX = Y$$

wobei X und Y Vektoren sind.

Wir zeigen, daß wir damit eine lineare Abbildung definiert haben:

L1: $F(\overline{x}+\overline{y}) = F(\overline{x})+F(\overline{y})$

$$A(X+Y) = A * \begin{bmatrix} x1+y1 \\ . \\ . \\ . \\ xn+yn \end{bmatrix} =$$

$$\begin{bmatrix} a11(x1+y1)+\dots+a1n(xn+yn) \\ . \\ . \\ . \\ am1(x1+y1)+\dots+amn)xn+yn) \end{bmatrix} =$$

$$\begin{bmatrix} a11x1+a11y1+\dots+a1nxn+a1nyn \\ . \\ . \\ . \\ am1x1+am1y1+\dots+amnxn+amnyn \end{bmatrix} = AX + AY$$

L2: $KF(\overline{x}) = F(K\overline{x})$

$K(AX) = (KA)X = A(KX)$

Wir können nun aber Gleichungssysteme, wie wir sie vorher behandelt haben, vollständig durch Matrizen darstellen, denn ausgeschrieben sieht das so aus:

$$AX+b = 0$$

$$\Leftrightarrow \begin{bmatrix} a11x1+\ldots a1nxn \\ \cdot \\ \cdot \\ \cdot \\ am1x1+\ldots amnxn \end{bmatrix} + \begin{bmatrix} b1 \\ b2 \\ \cdot \\ \cdot \\ bn \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ 0 \end{bmatrix}$$

Für einen Spezialfall der Gleichungen können wir nun aber dieses Gleichungssystem mit Hilfe von Matrizen lösen.
Ist nämlich A eine n-reihige quadratische Matrix, so gilt folgendes:

$$\begin{aligned} & AX+b = 0 \\ \Leftrightarrow \quad & AX = -b \\ \Leftrightarrow \quad & A\,AX = A(-b) \\ \Leftrightarrow \quad & EnX = A(-b) \\ \Leftrightarrow \quad & X = A(-b) \end{aligned}$$

Schreiben wir Y anstelle von -b, so erhalten wir

$$AX = Y \quad \Leftrightarrow \quad X = A\,Y$$

Wir erhalten damit eine eindeutig bestimmte Lösung, d.h. einen Punkt. Dazu muß allerdings der Rang von A gleich n sein, wie wir bereits gesehen haben. Dies macht verständlich warum die Forderung für die Invertierbarkeit einer Matrix lautete: Quadratisch und vom Rang n.
Wir können also jetzt ganz einfach die Lösung eines linearen Gleichungssystems bestimmen. Wir benötigen hierzu lediglich die Inverse Matrix A^{-1}. Um sie zu berechnen gibt es zwei prinzipielle Methoden:

(1) Aus dem unbestimmten Ansatz AX = En ermitteln wir mit Hilfe des Gauß'schen Eliminationsverfahrens die Form

$$X = A^{-1}\,En = A^{-1}$$

Diese Methode wird vor allem bei Matrizen mit RgA>4 angewandt und ist für uns praktisch bedeutungslos.
(2) Es gilt:

$$A^{-1} = \frac{1}{\det A} \begin{bmatrix} A11 & A12 & \ldots & A1n \\ A21 & A22 & \ldots & A2n \\ \cdot & & & \cdot \\ \cdot & & & \cdot \\ Am1 & Am2 & \ldots & Amn \end{bmatrix}^t$$

wobei A11 die zu a11 gehörige Adjunkte ist usw.. Die zur Komponente aij gehörige Adjunkte berechnet sich wie folgt:
Man streicht die i-te Zeile und die j-te Spalte von A und berechnet die mit dem Faktor (-1) versehene Restdeterminante. Da wir Determinanten nur bis zu 3-reihigen Determinanten sehr einfach berechnen können, greift man bei größeren Matrizen entweder zu Methode (1) oder zu anderen numerischen Verfahren (vgl. Herrmann,Weber:Mathematik auf dem Commodore 64,IWT-Verlag,1984)

A21 = ?

$$A = \begin{bmatrix} a11 & a12 & a13 \\ a21 & a22 & a23 \\ a31 & a32 & a33 \end{bmatrix}$$

Es bleibt zu berechnen:

$$(-1)^{(2+1)} \det \begin{bmatrix} a12 & a13 \\ a32 & a33 \end{bmatrix} = (-1)(a11a33-a32a13) = A21$$

Beispiel:

$$\begin{bmatrix} 3 & 2 & 1 \\ 1 & 0 & 2 \\ 4 & 1 & 3 \end{bmatrix}^{-1} = \frac{1}{5} \begin{bmatrix} -2 & 5 & 1 \\ -5 & 5 & 5 \\ 4 & -5 & -2 \end{bmatrix}^{t} = \frac{1}{5} \begin{bmatrix} -1 & -5 & 4 \\ 5 & 5 & -5 \\ 1 & 5 & -2 \end{bmatrix}$$

Wäre jetzt z.B. die Lösung des Gleichungssystems

$$\begin{aligned} 3\,x1 + 2\,x2 + x3 &= 1 \\ x1 + 2\,x3 &= 2 \\ 4\,x1 + x2 + 3\,x3 &= 3 \end{aligned}$$

gesucht, so würden wir die Lösung folgendermaßen bestimmen:

$$\begin{bmatrix} 3 & 2 & 1 \\ 1 & 0 & 2 \\ 4 & 1 & 3 \end{bmatrix} \begin{bmatrix} x1 \\ x2 \\ x3 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

$$\begin{bmatrix} x1 \\ x2 \\ x3 \end{bmatrix} = \frac{1}{5} \begin{bmatrix} -2 & 5 & 1 \\ -5 & 5 & 5 \\ 4 & -5 & -2 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 11/5 \\ 4 \\ 0 \end{bmatrix}$$

Wir können lineare Gleichungssysteme mit Hilfe von Matrizen sehr übersichtlich und einfach darstellen.

Dies mag bisher den Eindruck erweckt haben, daß wir uns in erster Linie mit dem Lösen von linearen Gleichungssystemen beschäftigen. Eigentlich ist dies aber gar nicht meine Absicht. Das bisher Gesagte diente in erster Linie zur Veranschaulichung einfacher Matrizenoperationen. Was für uns sehr viel wichtiger ist, ist daß wir durch die Matrizen ein Hilfsmittel an die Hand bekommen haben, das es uns ermöglicht Funktionen von Vektoren zu definieren. Bisher hatten wir uns lediglich damit beschäftigt mit Vektoren algebraische Umformungen vorzunehmen, sprich wir haben addiert und multipliziert. Mit Hilfe von Matrizen können wir jetzt ganze Funktionen definieren. Dies wird an folgender Analogie deutlich:

$y = f(x)$ $\qquad$ $Y = F(X)$

$y = 2x^2 + 3x + 1$ $\qquad$ $Y = AX$

Um uns die Funktion der Matrix deutlich zu machen betrachten wir das folgende Beispiel:

$$A = \begin{vmatrix} 2 & 1 & 3 \\ 4 & 3 & 1 \\ 0 & 3 & -2 \end{vmatrix}$$

Es sei $Y = AX$

$$\begin{bmatrix} y1 \\ y2 \\ y3 \end{bmatrix} = \begin{bmatrix} 2 & 1 & 3 \\ 4 & 3 & 1 \\ 0 & 3 & -2 \end{bmatrix} \begin{bmatrix} x1 \\ x2 \\ x3 \end{bmatrix}$$

Wir setzen die Standardbasisvektoren $\overline{e}1, \overline{e}2, \overline{e}3$ in diese Gleichung ein:

$$\begin{bmatrix} 2 & 1 & 3 \\ 4 & 3 & 1 \\ 0 & 3 & -2 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 2 & 1 & 3 \\ 4 & 3 & 1 \\ 0 & 3 & -2 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \\ 3 \end{bmatrix}$$

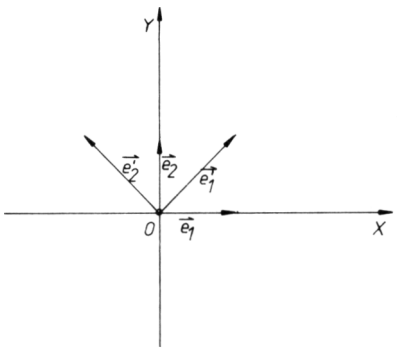
$$\begin{bmatrix} 2 & 1 & 3 \\ 4 & 3 & 1 \\ 0 & 3 & -2 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \\ -2 \end{bmatrix}$$

Wie wir sehen sind die Spaltenvektoren die Bilder der Basisvektoren.
Die Bilder der Basisvektoren sind gleichzeitig eine Basis des Bildraumes, der durch $Y=AX$ definiert ist. Dies folgt unmittelbar aus der

Linearität der Matrizenoperationen:

$$A(a_1e_1+a_2e_2+\ldots+a_ne_n)=A(a_1e_1)+A(a_2e_2)+\ldots+A(a_ne_n)=$$
$$=a_1A(e_1)+a_2A(e_2)+\ldots+a_nA(e_n)=a_1e_1'+a_2e_2'+\ldots+a_ne_n'$$

Wir wollen die Nützlichkeit der Matrizen an einem ausführlichen Beispiel veranschaulichen. (Es ist empfehlenswert sich dieses Beipiel sehr genau zu überlegen, denn es ist die Grundlage für fast alle Matrizenoperationen, von denen wir im Teil 2 Gebrauch machen werden.)



Gegeben sei der R^2, sowie die beiden Basen $(\bar{e}1,\bar{e}2)$ und $(\bar{e}1',\bar{e}2')$.
Es soll ferner gelten:

$$\bar{e}1 = \begin{bmatrix}1\\0\end{bmatrix};\ \bar{e}2 = \begin{bmatrix}0\\1\end{bmatrix}$$

$$\bar{e}1' = \begin{bmatrix}1\\1\end{bmatrix};\ \bar{e}2' = \begin{bmatrix}-1\\1\end{bmatrix}$$

Wir wollen im folgenden das System, das von den beiden Einheitsvektoren $\bar{e}1$ und $\bar{e}2$ aufgespannt wird als S, das der Vektoren $\bar{e}1'$ und $\bar{e}2'$ als S' bezeichnen.
Im System S' haben $\bar{e}1'$ und $\bar{e}2'$ die Koordinaten

$$\bar{e}1' = \begin{bmatrix}1\\0\end{bmatrix};\ \bar{e}2' = \begin{bmatrix}0\\1\end{bmatrix}$$

Ihre Koordinatendarstellung im alten System S erhalten wir, wie man leicht sieht, durch:

$$\overline{e}_{S}1' = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$\overline{e}_{S}2' = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

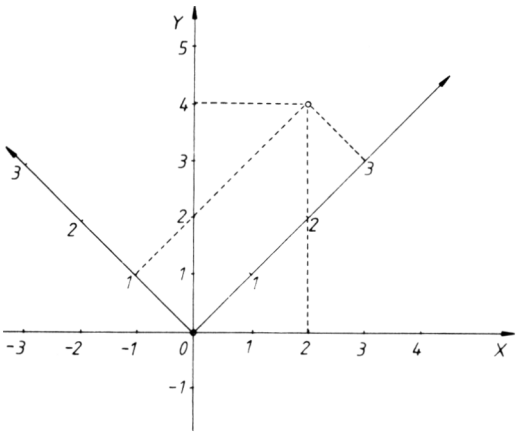
Wenn wir die Koordinaten eines Punktes im System S' kennen, so können wir mit Hilfe der Matrix A

$$A = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$$

seine Koordinaten im System S bestimmen.
Sei z.B. der Punkt $\overline{P}'$ (3/1) im System S' gegeben. Mit

$$\overline{P} = A\overline{P}' = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \end{bmatrix}$$

erhalten wir die Koordinaten (2/4) im alten System S.
Von der Richtigkeit überzeugen wir uns anhand einer Skizze.



Es gilt also:

$$\overline{x} = A\overline{x}' \quad (*)$$

wobei A spaltenweise aus den Koordinaten der

Basisvektoren $\overline{e1}'$,$\overline{e2}'$,... bezüglich der Basis von System S bestehen.
Aus Gleichung (*) sehen wir, daß wir diese Gleichung auch umgekehrt durchführen können, denn es gilt:

$$\overline{X} = A\overline{X}'$$

$$\Leftrightarrow A^{-1}\overline{X} = \overline{X}'$$

Wir bestimmen A^{-1} am einfachsten nach Methode 2

$$\begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}^{-1} = \frac{1}{2}\begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}^{t} = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}$$

Die Richtigkeit prüfen wir mittels

$$AA^{-1} = En$$

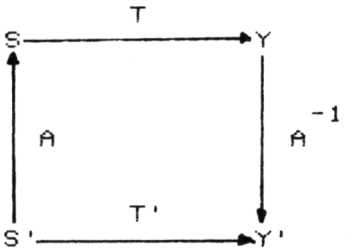
$$\frac{1}{2}\begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Wollen wir nun die Koordinaten des Punktes $\overline{P}$=(2/4) im System S' bestimmen, so gilt:

$$\overline{P}' = A^{-1}\overline{P}$$

$$\overline{P}' = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}\begin{bmatrix} 2 \\ 4 \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \end{bmatrix}$$

Wir können also sehr einfach mit Hilfe von A und A^{-1} vom System S ins System S' und umgekehrt rechnen.
Nehmen wir nun an, wir hätten eine Abbildung T in S gegeben, mit Y=TX. Es interessiert uns nun wie die Matrix T' aussehen muß, die die selbe Funktion in S' beschreibt, wie sie T in S beschreibt.
Dazu betrachten wir folgendes Diagramm:



Bei gegebenem Punkt $\overline{X}'$ müßen wir zuerst von S' nach S zurückrechnen, dann T ausführen und danach wieder von S nach S' umrechnen.
Es gilt also: $\overline{Y}' = A^{-1}TA\overline{X}'$
Damit haben wir eine Formel erhalten, mit deren Hilfe wir eine Abbildung von einem System in ein anderes transformieren können.

Beispiele für einfache Matrizenoperationen

Wir wollen uns damit jetzt in ein paar Beispielen vertraut machen. Ich möchte Ihnen empfehlen die Beispiele selbst durchzurechnen, und die Lösungen erst später zu betrachten.

Aufgabe(1): Es sei $(\bar{e}1,\bar{e}2)$ die Standardbasis des R^2.
a) Warum wird durch die Gleichungen

$$\bar{e}1' = 2\bar{e}1+\bar{e}2 \quad ; \quad \bar{e}2' = \bar{e}1+\bar{e}2$$

eine neue Basis $(\bar{e}1',\bar{e}2')$ definiert?

b) Man berechne die Koordinaten (x1'/x2') eines Vektors aus seinen alten Koordinaten (x1/x2) und umgekehrt.

c) Man zeichne in ein Koordinatensystem die Punkte

$$\begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}0\\1\end{bmatrix}, \begin{bmatrix}1\\1\end{bmatrix}$$

und die Geraden

$$K\begin{bmatrix}1\\-1\end{bmatrix} ; \begin{bmatrix}1\\0\end{bmatrix}+K\begin{bmatrix}1\\2\end{bmatrix} \quad ;K \text{ aus } R$$

ein,wobei diese Angaben einmal als alte, zum anderen als neue Koordinaten aufgefaßt werden sollen.

d) Durch welche Gleichungen werden die Koordinatenachsen in beiden Koordinatensystemen beschrieben?

LÖSUNG

a)
$$\bar{e}1 = \begin{bmatrix}1\\0\end{bmatrix}; \quad \bar{e}2 = \begin{bmatrix}0\\1\end{bmatrix}$$

$$\bar{e}1' = \begin{bmatrix}2\\1\end{bmatrix}; \quad \bar{e}2' = \begin{bmatrix}-1\\1\end{bmatrix}$$

$(\bar{e}1',\bar{e}2')$ bilden eine Basis des R^2, da sie linear unabhängig sind.

b)Die Spaltenvektoren der Transformationsmatrix sind die Bilder der Basisvektoren.Es gilt also

$$A = \begin{bmatrix} 2 & -1 \\ 1 & 1 \end{bmatrix}$$

$$\overline{X} = A\overline{X}'$$

$$A^{-1} = \frac{1}{3}\begin{bmatrix} 1 & -1 \\ 1 & 2 \end{bmatrix}^t = \frac{1}{3}\begin{bmatrix} 1 & 1 \\ -1 & 2 \end{bmatrix}$$

Probe:

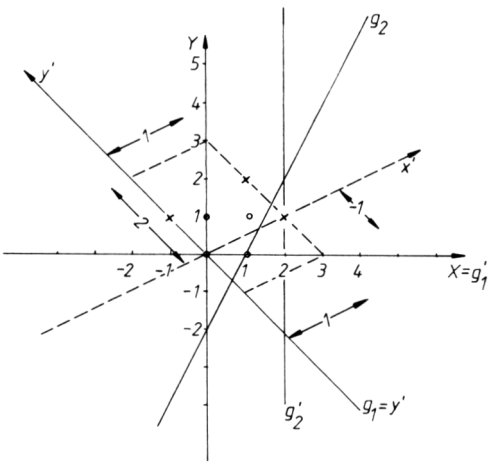
$$\frac{1}{3}\begin{bmatrix} 2 & -1 \\ 1 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 1 \\ -1 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\overline{X}' = A^{-1}\overline{X}$$

Damit gilt also

$$\begin{bmatrix} x1 \\ x2 \end{bmatrix} = \begin{bmatrix} 2 & -1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x1' \\ x2' \end{bmatrix}$$

$$\begin{bmatrix} x1' \\ x2' \end{bmatrix} = \frac{1}{3}\begin{bmatrix} 1 & 1 \\ -1 & 2 \end{bmatrix} \begin{bmatrix} x1 \\ x2 \end{bmatrix}$$



Wir wählen als Koordinatensystem das System S.
(o) alte Koordinaten :

$$g1: \overline{X} = K \begin{bmatrix} 1 \\ -1 \end{bmatrix} \; ; \; g2: \overline{X} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} + K \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

(+) neue Koordinaten :

Richtungsvektoren der Koordinatenachsen $\overline{X}', \overline{Y}'$:

$$\begin{bmatrix} 2 & -1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} 2 & -1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} 2 & -1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

$$g1': \quad \begin{bmatrix} 2 & -1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \end{bmatrix} = \begin{bmatrix} 3 \\ 0 \end{bmatrix} \qquad g1': \overline{X} = K \begin{bmatrix} 3 \\ 0 \end{bmatrix}$$

$$g2': \quad \begin{bmatrix} 2 & -1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \end{bmatrix} \qquad g2': \overline{X} = \begin{bmatrix} 2 \\ 1 \end{bmatrix} + K \begin{bmatrix} 0 \\ 3 \end{bmatrix}$$

d) Koordinatenachsen in S:

$$X: K \begin{bmatrix} 1 \\ 0 \end{bmatrix} \; ; \; Y: K \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$X': K \begin{bmatrix} 2 \\ 1 \end{bmatrix} \; ; \; Y': K \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

Koordinatenachsen in S':

$$\frac{1}{3} \begin{bmatrix} 1 & 1 \\ -1 & 2 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \frac{1}{3} \begin{bmatrix} 1 \\ -1 \end{bmatrix} = \begin{bmatrix} 1/3 \\ -1/3 \end{bmatrix}$$

$$X: K \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

$$\frac{1}{3}\begin{bmatrix}1 & 1\\ -1 & 2\end{bmatrix}\begin{bmatrix}0\\ 1\end{bmatrix} = \frac{1}{3}\begin{bmatrix}1\\ 2\end{bmatrix} = \begin{bmatrix}1/3\\ 2/3\end{bmatrix}$$

$$\overline{Y}: \; k\begin{bmatrix}1\\ 2\end{bmatrix}$$

$$\overline{X}': \; k\begin{bmatrix}1\\ 0\end{bmatrix}; \quad \overline{Y}': \; k\begin{bmatrix}0\\ 1\end{bmatrix}$$

Aufgabe(2): Durch $\overline{X}' = T\overline{X} =$

$$\begin{bmatrix}1 & 1 & 0\\ -1 & 0 & 0\\ -1 & -1 & 1\end{bmatrix}\overline{X}$$

wird eine Koordinatentransformation im R^3 ausgeführt.
a) Man berechne die Gleichung $\overline{X}=S\overline{X}'$ für die umgekehrte Transformation.
b) Man zeige,daß die Ebene $x1+x2=0$ im gestrichenen System zu einer Koordinatenebene wird und die Gerade

$$g:\overline{X} = k\begin{bmatrix}0\\ 1\\ 1\end{bmatrix}$$

zu einer Koordinatenachse.

LÖSUNG:

a)

$$S = T^{-1} = {}^{t}\begin{bmatrix}0 & 1 & 1\\ -1 & 1 & 0\\ 0 & 0 & 1\end{bmatrix} = \begin{bmatrix}0 & -1 & 0\\ 1 & 1 & 0\\ 1 & 0 & 1\end{bmatrix}$$

Probe:

$$\begin{bmatrix}1 & 1 & 0\\ -1 & 0 & 0\\ -1 & -1 & 1\end{bmatrix} * \begin{bmatrix}0 & -1 & 0\\ 1 & 1 & 0\\ 1 & 0 & 1\end{bmatrix} = \begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}$$

Damit ist die umgekehrte Transformation gefunden.

$$\overline{X} = \begin{bmatrix}0 & -1 & 0\\ 1 & 1 & 0\\ 1 & 0 & 1\end{bmatrix}\overline{X}'$$

b) E: x1+x2 = 0

Wir bestimmen die Gleichungen der Koordinatenachsen von S' in S

$$\begin{bmatrix} 0 & -1 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} \quad (*)$$

$$\begin{bmatrix} 0 & -1 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & -1 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Um die parameterfreien Gleichungen der Koordinatenebenen (x1x2-, x2x3-, x1x3-Ebene) zu erhalten, bestimmen wir die Kreuzprodukte der Koordinatenvektoren:

$$\begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} \times \begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \\ 1 \end{bmatrix} ; \quad \begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} ; \quad \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} \times \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

Wir finden daher, da der Ursprung (0/0/0) in allen 3 Ebenen enthalten sein soll und damit n0 jeweils 0 wird:

$$\begin{aligned} E1: &\ -x1 - x2 + x3 = 0 \\ E2: &\ \ x1 + x2 \ \ \ \ \ = 0 \\ E3: &\ \ x1 \ \ \ \ \ \ \ \ = 0 \end{aligned}$$

E2 ist die vorgegebene Ebene.
Die Gerade

$$g: X = k \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$$

ist offensichtlich genau die X1'-Achse, wie aus (*) ersichtlich ist.

Aufgabe(3): Es sei

$$A = \begin{bmatrix} \cos w & -\sin w \\ \sin w & \cos w \end{bmatrix} \quad \text{mit } w \text{ aus } R.$$

a) Man bestimme die Inverse Matrix von A.

b) Man zeige, daß durch

$$\overline{Y} = A\overline{X}$$

eine Drehung des Vektors $\overline{X}$ um den Ursprung um einen Winkel w bewirkt wird.
c) Man bestimme die Gleichung der Geraden

$$\begin{bmatrix}1\\1\end{bmatrix} + K\begin{bmatrix}1\\\sqrt{3}\end{bmatrix}$$

in einem um den Winkel $w = \pi/6$ (:= 30°) gedrehten System S'.

LÖSUNG:

a) $\det A = \cos^2 w + \sin^2 w = 1$ (Satz des Pythagoras)

$$A^{-1} = \begin{bmatrix}\cos w & -\sin w\\ \sin w & \cos w\end{bmatrix}^t = \begin{bmatrix}\cos w & \sin w\\ -\sin w & \cos w\end{bmatrix}$$

Es gilt also $A^{-1} = A$.
Matrizen, die diese Bedingung erfüllen bezeichnet man als orthogonale Matrizen. Für sie gilt, daß Matrizenoperationen mit ihnen kommutativ sind, d.h. es gilt (AB)X = (BA)X.

b)

$$\begin{bmatrix}y1\\y2\end{bmatrix} = \begin{bmatrix}\cos w & -\sin w\\ \sin w & \cos w\end{bmatrix}\begin{bmatrix}x1\\x2\end{bmatrix}$$

$$\begin{bmatrix}y1\\y2\end{bmatrix} = \begin{bmatrix}x1\cos w - x2\sin w\\ x1\sin w + x2\cos w\end{bmatrix}$$

Wir bestimmen mittels Skalarprodukt den Winkel v zwischen dem Vektor $\overline{X}$ und dem Vektor $\overline{Y}$.
Es gilt:

$$\cos v = \frac{\overline{X} \circ \overline{Y}}{|\overline{X}|\,|\overline{Y}|}$$

$$\Leftrightarrow \quad \cos v = \frac{\begin{bmatrix}x1\\x2\end{bmatrix} \circ \begin{bmatrix}x1\cos w - x2\sin w\\ x1\sin w + x2\cos w\end{bmatrix}}{|\overline{X}|\,|\overline{Y}|}$$

$$\Leftrightarrow \quad \cos v = \frac{x1^2\cos w - x1x2\cos w + x1x2\sin w + x2^2\cos w}{|\bar{X}|\,|\bar{Y}|}$$

$$\Leftrightarrow \quad \cos v = \frac{\cos w\,(x1^2+x2^2)}{|\bar{X}|\,|\bar{Y}|} =$$

$$= \frac{\cos w\ |\bar{X}|^2}{|\bar{X}|\,|\bar{Y}|} = \cos w\ \frac{|\bar{X}|}{|\bar{Y}|} \quad (*)$$

Für $|\bar{Y}|$ gilt aber:

$$|\bar{Y}|^2 = (x1\cos w - x2\sin w)^2 + (x1\sin w + x2\cos w)^2 =$$

$$= x1^2\cos^2 w - 2x1x2\cos w\sin w + x2^2\sin^2 w +$$
$$x1^2\sin^2 w + 2x1x2\cos w\sin w + x2^2\cos^2 w =$$
$$= x1^2(\cos^2 w + \sin^2 w) + x2^2(\cos^2 w + \sin^2 w) = x1^2 + x2^2$$

Damit gilt also

$$|\bar{Y}| = \sqrt{x1^2+x2^2} = |\bar{X}|$$

und damit

$$\frac{|\bar{Y}|}{|\bar{X}|} = 1 \quad \Rightarrow \cos v = \cos w$$

$\Rightarrow v = w$, da nur Winkel $< 180^\circ$ in Frage kommen. Gleichung (*) ist damit immer erfüllt, womit gezeigt ist, daß A eine Drehung um den Winkel w bewirkt.

c) Wir rechnen vom System S in das System S' um. Es gilt:

$$\bar{X}' = A\,\bar{X}$$

Weiterhin gelten die Beziehungen:

$$\cos 30^\circ = \frac{1}{2}\sqrt{3} \qquad \sin 30^\circ = \frac{1}{2}$$

Wir können also A^{-1} schreiben:

$$A^{-1} = \begin{bmatrix} .5\sqrt{3} & .5 \\ -.5 & .5\sqrt{3} \end{bmatrix} = \frac{1}{2}\begin{bmatrix} \sqrt{3} & 1 \\ -1 & \sqrt{3} \end{bmatrix}$$

Somit finden wir:

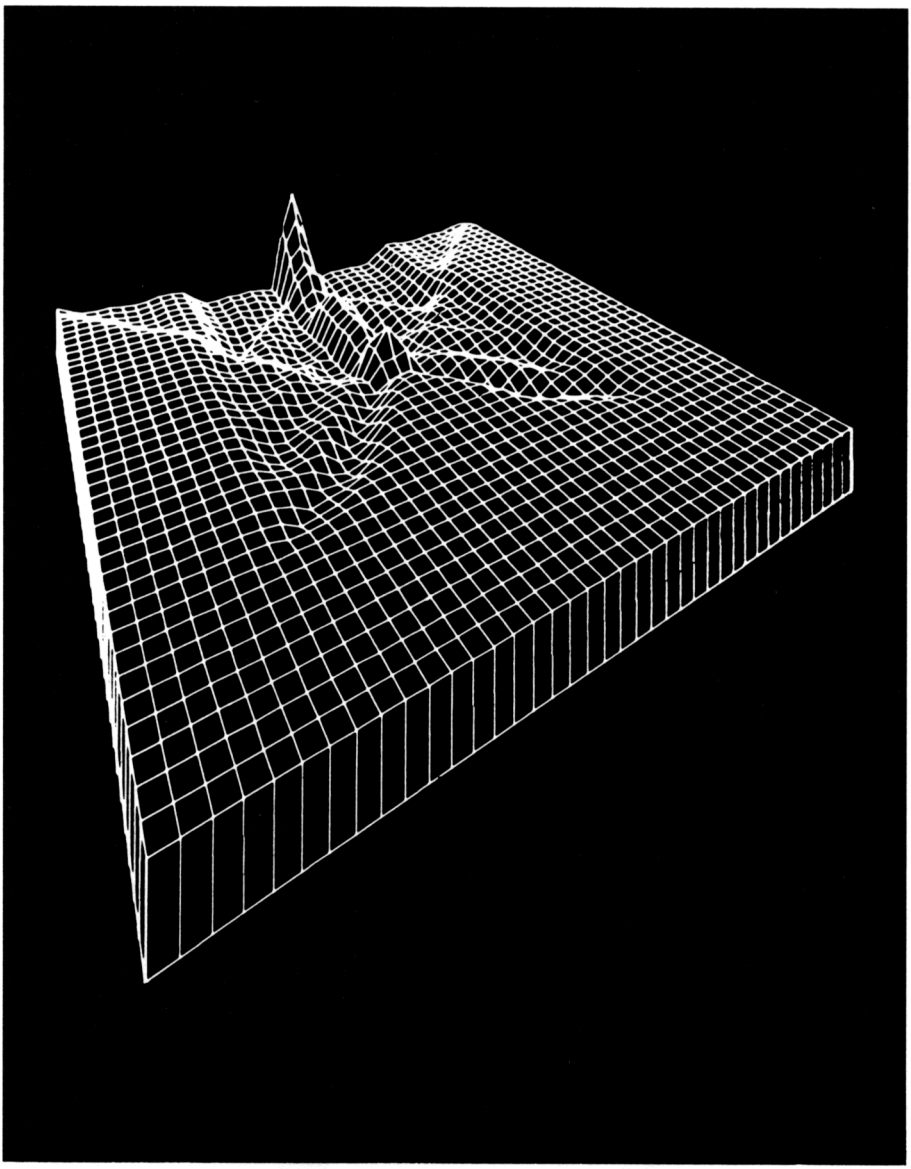
$$\frac{1}{2}\begin{bmatrix}\sqrt{3} & 1\\ -1 & \sqrt{3}\end{bmatrix}\begin{bmatrix}1\\ 1\end{bmatrix}=\begin{bmatrix}(\sqrt{3}+1)/2\\ (\sqrt{3}-1)/2\end{bmatrix}$$

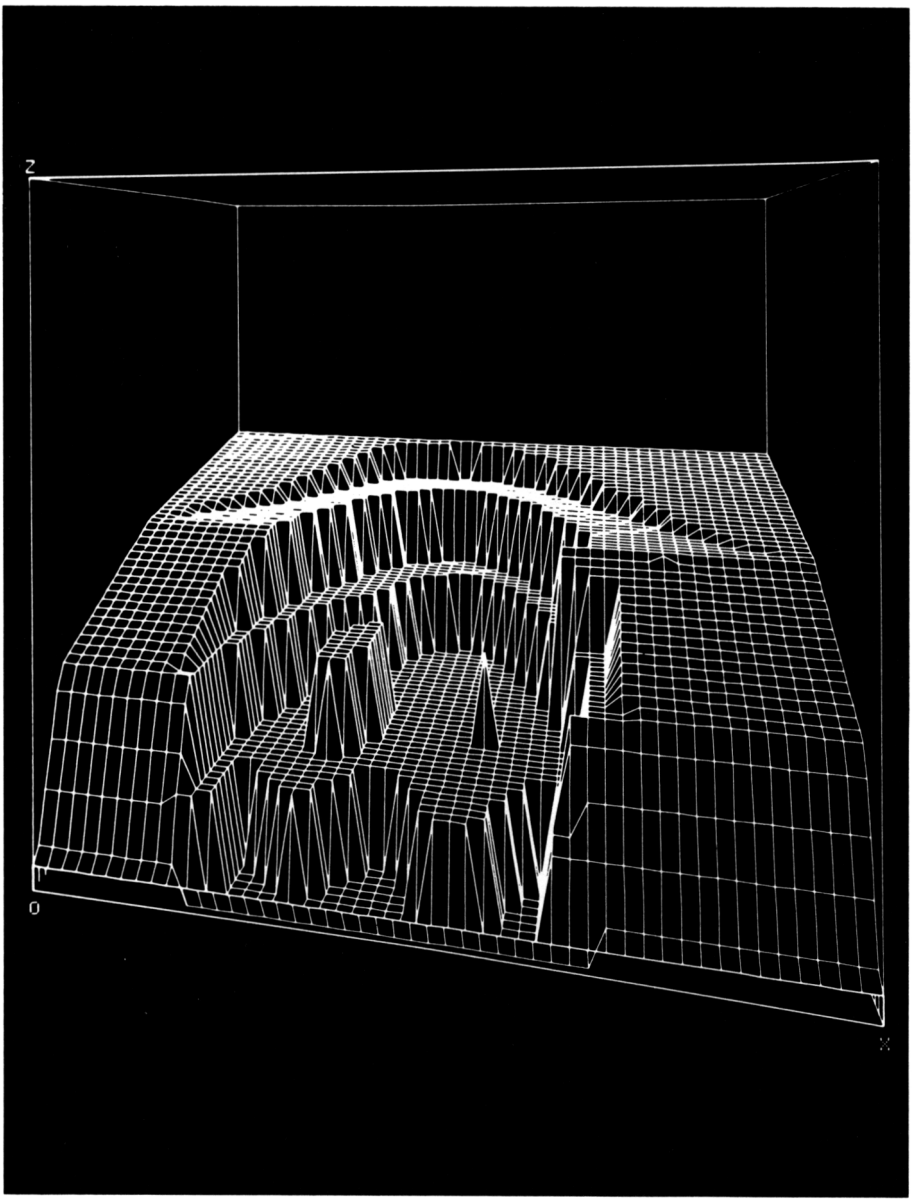
$$\frac{1}{2}\begin{bmatrix}\sqrt{3} & 1\\ -1 & \sqrt{3}\end{bmatrix}\begin{bmatrix}1\\ \sqrt{3}\end{bmatrix}=\begin{bmatrix}\sqrt{3}\\ 1\end{bmatrix}$$

Die Gleichung der Geraden in S' lautet also:

$$\overline{X}=\begin{bmatrix}\frac{\sqrt{3}+1}{2}\\ \frac{\sqrt{3}-1}{2}\end{bmatrix}+K\begin{bmatrix}\sqrt{3}\\ 1\end{bmatrix}$$

Wir haben damit nun alle Grundlagen besprochen, die wir für die Erstellung von Grafiken benötigen. Daß man diese Kenntnisse aber durchaus auch anders nutzen kann werden sie am Beispiel der Simulation des 3-Koerperproblems sehen.




Z
0
X

TEIL 2

Wir wollen uns jetzt mit der Anwendung der Vektorrechnung auf graphische Probleme befassen. Hierzu ein paar allgemeine Anmerkungen vorweg:

Die Programme sind für den hochaufgelösten Grafikbildschirm des C-64 geschrieben. Außer den speziellen Bildschirmpositionen und der Anordnung von X- und Y-Koordinatenachse des Bildschirms sind die Programme aber auch auf allen anderen grafikfähigen BASIC-Computern lauffähig wenn deren spezifische Eigenheiten berücksichtigt werden.

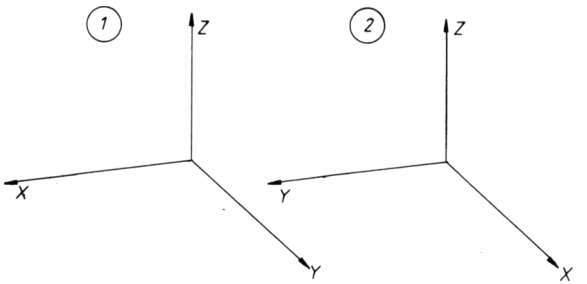
Aufbau des Grafikbildschirms

Der Grafikbildschirm des C-64 besitzt eine 320x200 Einzelpunktgrafik. Daneben ist es auch möglich den Bildschirm im Multicolour-Bitmap-Modus (Kurz MCB-Modus) zu betreiben. Wir beschränken uns hier auf den normalen HIRES-Betrieb, in dem nur Hintergrund- und eine Linienfarbe frei wählbar sind. Im MCB-Modus sind dagegen neben der Hintergrund- und einer Linienfarbe noch weitere 3 Linienfarben wählbar. Das Handicap dabei ist allerdings, daß die Auflösung dabei auf die Hälfte reduziert wird, also auf 160x200 Einzelpunkte schrumpft. Die Ursache hierfür liegt darin, daß 2 Bits für einen Punkt benötigt werden, da durch die 4 Möglichkeiten 00,01,10,11 der Doppelbitbelegung neben der Bildschirmposition gleichzeitig eine von 4 zur Verfügung stehenden Farben ausgewählt wird.
Um die Programme so kurz und übersichtlich wie möglich zu halten greifen wir in diesem Buch auf das Grafiksystem IWT SPRITE KOMFORT KIT zurück, da der C-64 zwar eine hervorragende hochaufgelöste Grafik, aber im Gegensatz zu anderen vergleichbaren Computern keine speziellen Maschinenbefehle zu ihrer Ansteuerung besitzt.Um es Ihnen zu ermöglichen die Programme entsprechend zu modifizieren, finden sie im Anhang sowohl eine Grafiksubroutine, die BASIC und Maschinensprache kombiniert verwendet und es Ihnen erlaubt auf die KIT-Befehle zu verzichten, als auch einen Auszug aus dem Befehlslisting, der ein Übertragen in andere Grafiksysteme ermöglicht.

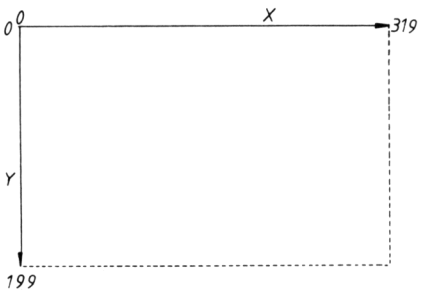
Eine Anmerkung zu den Listings: Die Programme sind mit den Kommentaren, die durch ****

gekennzeichnet sind nicht lauffähig.

Wir werden uns im folgenden häufig mit dreidimensionalen Problemen beschäftigen.Auch hierzu sei ein Wort vorweg bemerkt.
Es wird Ihnen vielleicht auffallen, daß in den dreidimensionalen Koordinatensystemen x- und y-Achse des öfteren in ihrer Reihenfolge vertauscht auftreten.



Fall (1) ist die mathematisch richtige Anordnung. Wir werden sie immer für rein mathematische Herleitungen verwenden.
Durch die Anordnung von x- und y-Achse auf dem HIRES-Bildschirm



dreht sich diese Konstellation für den Betrachter um (Fall (2)). Wir wählen deshalb die zweite Darstellung immer dann, wenn es sich um spezielle Probleme der Bildschirmgrafik handelt. Der Unterschied entsteht dadurch, daß wir die x-y-Ebene einmal von oben (1) und einmal von unten (2) sehen. Doch dies nur am Rande.

Wir setzen im folgenden voraus,das der Leser im Umgang mit den trigonometrischen Funktionen sin,cos und tan vertraut ist.

Geometrische Grundoperationen im R^2 und R^3

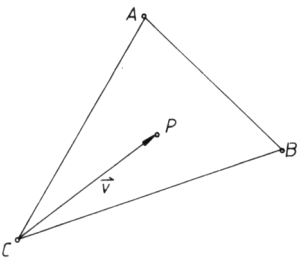
Wir werden uns jetzt mit geometrischen Grundoperationen (Verschieben, Achsenspiegeln, Punktspiegeln, Drehen) im R^2 und R^3 befassen. Wir werden nach Möglichkeit beide Fälle parallel behandeln, da die Unterschiede nicht sehr groß sind. Interessant sind in erster Linie allerdings nur die zweidimensionalen Operationen.

Verschiebung

Wie wir bereits in Teil 1 gesehen haben, können wir eine Figur bzw. einen Körper, der durch endlich viele Ortsvektoren beschrieben werden kann, mit Hilfe eines Verschiebungsvektors an jede beliebige Stelle verschieben. Wir wollen dies am Beispiel eines Dreiecks im R^2 einmal durchführen. Im R^3 läuft die entsprechende Operation analog.

R^2:

Gegeben sei das Dreieck ABC sowie der Punkt P.



Wir wollen das Dreieck so verschieben, daß C die neue Darstellung C'=P hat. Damit können wir den Verschiebungsvektor $\overline{v}$ wie folgt bestimmen.
Es gilt:

$$\overline{C}+\overline{v} = \overline{C}'$$

$$\Leftrightarrow \quad \overline{v} = \overline{C}'-\overline{C} = \overline{P}-\overline{C}$$

Wir erhalten dann das neue Dreieck A'B'C' durch Addition des Verschiebungsvektors $\overline{v}$ zu jedem einzelnen Ortsvektor A,B,C.
Damit erhalten wir:

$$\overline{A}' = \overline{A}+\overline{v} = \overline{A}+\overline{P}-\overline{C}$$
$$\overline{B}' = \overline{B}+\overline{v} = \overline{B}+\overline{P}-\overline{C}$$
$$\overline{C}' = \overline{P}$$

Den Computer lassen wir die Operation völlig analog für die einzelnen Komponenten des Vektors berechnen:
Sei z.B.

$$\overline{A} = \begin{bmatrix} a1 \\ a2 \end{bmatrix} \text{ und } \overline{v} = \begin{bmatrix} v1 \\ v2 \end{bmatrix} = \begin{bmatrix} p1-c1 \\ p2-c2 \end{bmatrix},$$

dann erhalten wir die neue Position durch

$$\overline{A}' = \overline{A}+\overline{v} = \begin{bmatrix} a1 \\ a2 \end{bmatrix} + \begin{bmatrix} v1 \\ v2 \end{bmatrix} = \begin{bmatrix} a1+v1 \\ a2+v2 \end{bmatrix} = \begin{bmatrix} a1' \\ a2' \end{bmatrix}$$

und damit den Programmabschnitt:

```
A1' = (A1+V1) = A1+P1-C1
A2' = (A2+V2) = A2+P2-C2
```

Spiegelungen

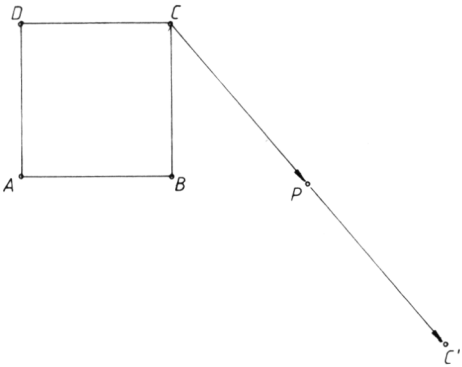
Auch hier laufen die Operationen im R^2 und R^3 analog ab. Es ist allerdings zu beachten, daß wir im R^2 nur zwischen Punkt- und Achsenspiegelung zu unterscheiden haben. Als analoge Operationen sind zu sehen:

R^2	R^3
Punktspiegelung	Punktspiegelung
Achsenspiegelung	Ebenenspiegelung

Die Achsenspiegelung im R^3 stellt eine Mischform aus Achsen- und Punktspiegelung im R^2 dar und wir werden sie hier nicht näher behandeln.

Punktspiegelungen

Wir betrachten als Beispiel das Quadrat ABCD im R^2, das am Punkt P gespiegelt werden soll:



Wir erhalten die neuen Koordinaten z.B. des Punktes C folgendermaßen:
Die Punkte C und P legen eine Gerade fest. Dabei ist der Vektor $\overline{CP} = \overline{P}-\overline{C}$ ein Richtungsvektor der Geraden und P ein Trägerpunkt. Der Punkt C' ist dadurch charakterisiert, daß er auf der Geraden durch C und P liegt und von P ebensoweit entfernt ist wie C.
Damit erhalten wir C' aber ganz einfach durch:

$$\overline{C}' = \overline{P}+\overline{CP} = \overline{P}+\overline{P}-\overline{C} = 2\overline{P}-\overline{C}$$

In Koordinatenschreibweise:

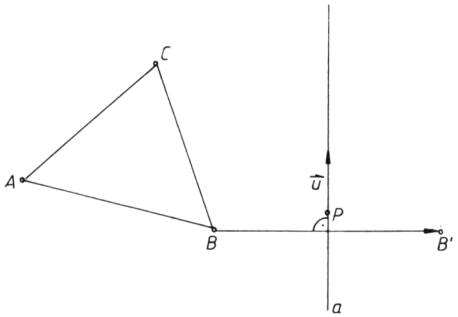
$$\overline{C}' = \begin{bmatrix} c1' \\ c2' \end{bmatrix} = \begin{bmatrix} 2p1-c1 \\ 2p2-c2 \end{bmatrix}$$

Allgemein also:

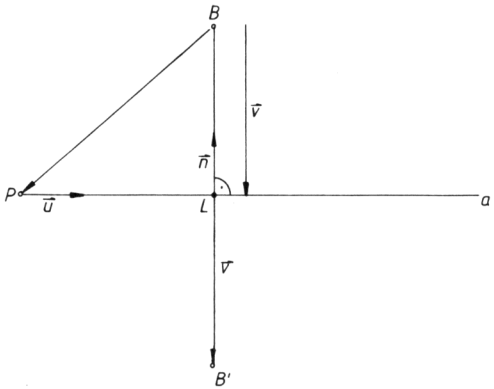
$$\begin{aligned} \overline{A}' &= 2\overline{P}-\overline{A} \\ \overline{B}' &= 2\overline{P}-\overline{B} \\ \overline{C}' &= 2\overline{P}-\overline{C} \\ \overline{D}' &= 2\overline{P}-\overline{D} \end{aligned}$$

Achsenspiegelung im R^2 und Ebenenspiegelung im R^3

Wir betrachten das Dreieck ABC



Es soll an der Achse a: $\overline{X} = \overline{P} + k\overline{u}$ gespiegelt werden. Um z.B. die neuen Koordinaten B' von B zu erhalten, gehen wir wie folgt vor:



Ist der Richtungsvektor $\overline{u}$ durch

$$\overline{u} = \begin{bmatrix} u1 \\ u2 \end{bmatrix}$$

gegeben, so erhalten wir einen Normalenvektor $\overline{n}$ von $\overline{a}$ durch

$$\overline{n} = \begin{bmatrix} -u1 \\ u2 \end{bmatrix}$$

da gilt $\overline{n} \circ \overline{u} = \overline{0}$.
Es ist außerdem keine Schwierigkeit den Vektor $\overline{BP} = \overline{P} - \overline{B}$ zu

$$\overline{BP} = \begin{bmatrix} p1 \\ p2 \end{bmatrix} - \begin{bmatrix} b1 \\ b2 \end{bmatrix} = \begin{bmatrix} p1-b1 \\ p2-b2 \end{bmatrix}$$

zu bestimmen.
Durch Projektion des Vektors $\overline{BP}$ in den Vektor $\overline{n}$ erhalten wir den Vektor $\overline{v}$, für den folgende Gleichung gilt:

$$\overline{v} = \overline{BL} = \overline{L} - \overline{B}$$

Für $\overline{B}'$ gilt die Beziehung:

$$\overline{B}' = \overline{B} + 2\overline{BL} = \overline{B} + 2\overline{v}$$

$\overline{v}$ können wir leicht durch die Projektionsformel (vgl. Teil 1) bestimmen. Es gilt damit:

$$\overline{v} = \frac{\overline{BP} \circ \overline{n}}{|\overline{n}|} \overline{n} = \frac{\begin{bmatrix} p1-b1 \\ p2-b2 \end{bmatrix} \circ \begin{bmatrix} -u2 \\ u1 \end{bmatrix}}{\sqrt{u1^2+u2^2}} \begin{bmatrix} -u1 \\ u2 \end{bmatrix}$$

Wir erhalten damit:

$$\overline{v} = \begin{bmatrix} v1 \\ v2 \end{bmatrix} = \frac{(p1-b1)(-u2)+(p2-b2)(u1)}{\sqrt{u1^2+u2^2}} \begin{bmatrix} -u2 \\ u1 \end{bmatrix}$$

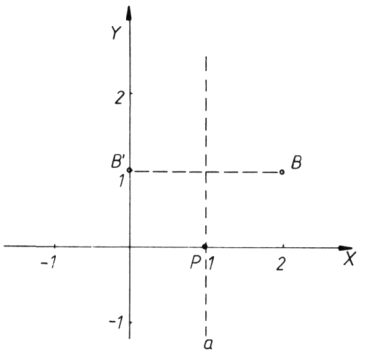
Für $\overline{B}'$ gilt demnach folgende Beziehung:

$$\overline{B}' = \overline{B} + 2\overline{v} = \begin{bmatrix} b1 \\ b2 \end{bmatrix} + \frac{212u1-211u2}{\sqrt{u1^2+u2^2}} \begin{bmatrix} -u2 \\ u1 \end{bmatrix}$$

wobei $\overline{l} = \begin{bmatrix} l1 \\ l2 \end{bmatrix} = \begin{bmatrix} p1-b1 \\ p1-b1 \end{bmatrix}$ ist.

Hierzu ein Beispiel:

Sei $\overline{B} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$ und a: $\overline{X} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} + K \begin{bmatrix} 0 \\ 1 \end{bmatrix}$



Wir wissen, daß $\overline{B}'$ die Koordinaten (0/1) besitzen muß und prüfen das nun rechnerisch nach.

$$\overline{l} = \begin{bmatrix} l1 \\ l2 \end{bmatrix} = \begin{bmatrix} p1-b1 \\ p2-b2 \end{bmatrix} = \begin{bmatrix} 1-2 \\ 0-1 \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$$

$$\overline{B}' = \begin{bmatrix} 2 \\ 1 \end{bmatrix} + \frac{2(-1)0 \;\; -2(-1)1}{1} \begin{bmatrix} -1 \\ 0 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix} + \begin{bmatrix} -2 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Die Ebenenspiegelung im R^3

Dieser Abschnitt ist nur für den mathematisch interessierten Leser gedacht und ist für die Grafikbearbeitung praktisch von keiner Bedeutung. Wir können uns hier sehr kurz fassen:
Gegeben sei der Punkt B sowie die Ebene E.

Ist E in Parameterform gegeben, so bestimmen wir einen Normalenvektor durch $\overline{u} \times \overline{v} = \overline{n}$, ansonsten ist ein Normalenvektor sowieso durch die Koeffizienten n1,n2 und n3 gegeben (vgl. Teil 1).

Bei bekannten Normalenvektor gilt aber die selbe Formel wie im R^2 und damit

$$\overline{B}' = \overline{B} + 2 \frac{\overline{l} \circ \overline{n}}{|\overline{n}|} \overline{n}$$

oder anders geschrieben:

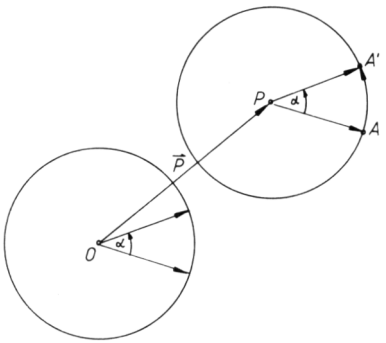
$$\overline{B}' = \begin{bmatrix} b1 \\ b2 \\ b3 \end{bmatrix} + \frac{2(l1n1+l2n2+l3n3)}{\sqrt{n1^2+n2^2+n3^2}} \begin{bmatrix} n1 \\ n2 \\ n3 \end{bmatrix}$$

Drehungen

Man kann Drehungen mit Hilfe von sog. Rotationsmatrizen beschreiben. Auf die Herleitung der Matrizenformel gehen wir noch im Abschnitt Rotationsmatrizen genauer ein. An dieser Stelle wollen wir deshalb nur den Umgang mit einer zweidimensionalen Rotationsmatrix besprechen.
Wie wir bereits in Teil 1 gesehen haben wird durch die Matrix

$$\begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix}$$

ein Ortsvektor $\overline{A}$ um den Winkel α um den Ursprung gedreht. Wollen wir einen Punkt A um einen beliebigen Punkt P um den Winkel α drehen, so gehen wir wie folgt vor.



Wir bestimmen den Vektor $\overline{PA}$.
Mit Hilfe der Rotationsmatrix Ra drehen wir den Vektor $\overline{PA}$ um den Winkel α und erhalten dadurch den Vektor $\overline{PA}'$. Durch Addition von $\overline{P}$ und $\overline{PA}'$ erhalten wir dann $\overline{PA}'$. Damit gilt allgemein die Formel:

$$\overline{A}' = \overline{P} + Ra(\overline{A}-\overline{P}) = \overline{P} + \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} (\overline{A}-\overline{P})$$

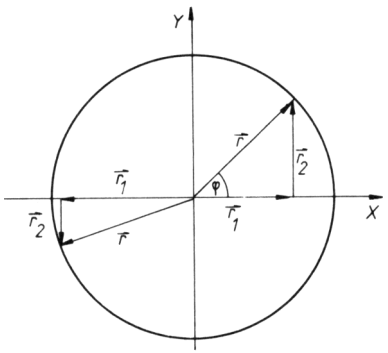
Grundstrukturen der Grafikprogrammierung:

Die Parameterform

Wir haben uns bisher nur darüber Gedanken gemacht, wie wir lineare Gebilde, z.B. Geraden, Dreiecke etc. vektoriell darstellen können. Dies ist aber genausogut mit Kreisen, Ellipsen, Spiralen, Kugeln etc. möglich. Wie wir gleich sehen werden, haben wir immer prinzipiell zwei Darstellungsarten, eine analytische und eine vektorielle, sog. parametrisierte, Form der Figurendefinition.

Kreise

Vielen ist vielleicht die Kreisgleichung $x^2 + y^2 = r^2$ bekannt. Es ist jedoch unschwer zu erkennen, daß sich diese Gleichung oder von dieser Gleichung abgeleitete Gleichungen sehr schlecht zur Programmierung eignen. Für eine einfachere Darstellung betrachten wir folgende Skizze:



Offensichtlich läßt sich der Radiusvektor $\overline{r}$, der zu einer beliebigen Position auf dem Kreis führt immer in zwei aufeinander senkrechte Komponenten $\overline{r}1$ und $\overline{r}2$ zerlegen.
Hierbei gilt für $\overline{r}1$ und $\overline{r}2$:

$$\overline{r}1 = |\overline{r}| \cos\phi \; \overline{x}$$
$$\overline{r}2 = |\overline{r}| \sin\phi \; \overline{y}$$

Damit haben wir also eine ganz einfache Möglichkeit gefunden einen Kreis zu Programmieren:
Es gilt, daß $|\overline{r}|\cos\phi$ der x-Komponente, $|\overline{r}|\sin\phi$

der y-Komponente von $\overline{r}$ entspricht und damit

$$\overline{r} = \begin{bmatrix} |\overline{r}|\cos\phi \\ |\overline{r}|\sin\phi \end{bmatrix} \text{ gilt.}$$

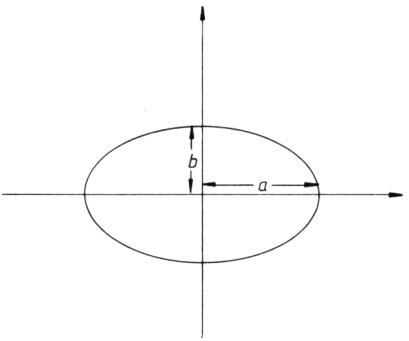
Diese Form der Kreisdarstellung bezeichnet man als parametrische Darstellung.
Wir lassen nun den Parameter alle Werte von 0 bis $2*\pi$ durchlaufen und plotten jeweils die Spitze des Radiusvektors. Dadurch erhalten wir einen Kreis. Das entsprechende Programm muß dann lauten:

```
z.B. R=10:MX=160:MY=100
     FOR I = 0 TO 2* STEP .1
     X = MX + R * COS(I)
     Y = MY + R * SIN(I)
    *PUNKT AN STELLE X,Y SETZEN*
     NEXT
```

Dabei sind MX und MY die Koordinaten des Mittelpunktes unseres Kreises. Wir können mit ihrer Hilfe den Mittelpunkt des Kreises, und damit den ganzen Kreis, vom Ursprung an eine beliebige Stelle am Bildschirm verschieben, denn es gilt die allgemeine Kreisgleichung

$$\overline{X} = \overline{M}+\overline{R}$$

Ellipsen



Der Kreis ist ein Spezialfall der Ellipse. Bei Ellipsen wird zwischen einer großen Halbachse a und einer kleinen Halbachse b unterschieden.
Man kann sich die Entstehung einer Ellipse bildlich dadurch vorstellen, daß man sich eine

Kreisscheibe im Raum um einen bestimmten Winkel gegenüber der Sichtlinie gedreht vorstellt. Was man sieht ist eine Ellipse. Die Ellipse wird deshalb mathematisch auch als affines Bild des Kreises bezeichnet.
Analog wie beim Kreis erhalten wir für die Parameterdarstellung der Ellipse die Form:

$$\vec{r} = \begin{bmatrix} a \sin\xi \\ b \cos\xi \end{bmatrix}$$

Während es beim Kreis egal ist, wie wir unser orthogonales Koordinatensystem legen, da immer der selbe Kreis entsteht, können wir bei den Ellipsen durch Drehung des Koordinatensystems eine Änderung ihrer Lage bewirken. Dazu kommen wir noch einmal auf die vektorielle Gleichung der Ellipse zurück. Es gilt:

$$\vec{r} = a\sin\xi\ \vec{e}1 + b\cos\xi\ \vec{e}2$$

wobei $\vec{e}1$ und $\vec{e}2$ die Einheitsvektoren unseres x-y-Koordinatensystems darstellen.
Wir können nun die Einheitsvektoren $\vec{e}1$ und $\vec{e}2$ durch zwei orthogonale Einheitsvektoren $\vec{e}1'$ und $\vec{e}2'$ ersetzen, die eine beliebige Lage in unserem ursprünglichen Koordinatensystem einnehmen.
Wir können dann die Koordinatendarstellung dieser Vektoren angeben. Es sei

$$\vec{e}1' = \begin{bmatrix} ax \\ ay \end{bmatrix}$$

Dann ist ein dazu orthogonaler Einheitsvektor gegeben durch

$$\vec{e}2' = \begin{bmatrix} -ay \\ ax \end{bmatrix} = \begin{bmatrix} bx \\ by \end{bmatrix}$$

Setzen wir jetzt in die Ellipsengleichung ein,so erhalten wir

$$\vec{r} = \begin{bmatrix} r1 \\ r2 \end{bmatrix} = a\sin\xi \begin{bmatrix} ax \\ ay \end{bmatrix} + b\cos\xi \begin{bmatrix} bx \\ by \end{bmatrix}$$

und damit die beiden Teile

$$r1 = a\sin\xi\ ax + b\cos\xi\ bx$$
$$r2 = a\sin\xi\ ay + b\cos\xi\ by$$

Dies ist die allgemeine Form der Ellipsengleichung. Wir erhalten sie allgemein durch die Koordinatentransformation:

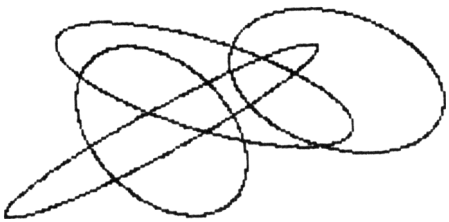
$$\begin{bmatrix} r1 \\ r1 \end{bmatrix} = \begin{bmatrix} ax & bx \\ ay & by \end{bmatrix} \begin{bmatrix} a\sin\xi \\ b\cos\xi \end{bmatrix}$$

Der Mittelpunkt der Ellipse liegt hierbei im Punkt (0/0). Durch Addition eines Ortsvektors

$$\overline{M} = \begin{bmatrix} mx \\ my \end{bmatrix}$$

erhalten wir dann wieder eine Verschiebung des Ellipsenmittelpunktes nach M.
Wie dies im Programm aussieht sehen sie unten. Sie brauchen dabei bei der Definition von $\overline{e1}$' und $\overline{e2}$' keine Rücksicht darauf zu nehmen, daß es sich um Einheitsvektoren handelt, da das Programm die Vektoren in den Zeilen 160-180 selbstständig in Einheitsvektoren verwandelt.

```
100 £0:£1:£L14,6
110 MX=160:MY=100
120 A=70:B=20
130 AX=2:AY=1
140 BX=-AY:BY=AX
150 :
160 Z=SQR(AX↑2+AY↑2)
170 AX=A*AX/Z:AY=A*AY/Z
180 BX=B*BX/Z:BY=B*BY/Z
190 FORI=0TO2**STEP.01
200 X=MX+SIN(I)*AX+COS(I)*BX
210 Y=MY+SIN(I)*AY+COS(I)*BY
220 £QX,Y
230 NEXT
240 GETA$:IFA$=""THEN250
250 £2
260 £2

READY.
```



Spiralen

Wir erhalten Spiralen direkt aus der Kreisdarstellung, wenn wir den Kreisradius proportional zu ε vergrößern bzw. verkleinern, z.B. durch

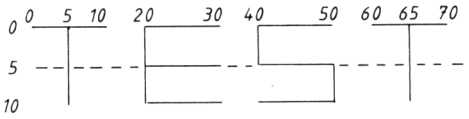
$$\overline{r} = \begin{bmatrix} \varepsilon\sin\varepsilon \\ \varepsilon\cos\varepsilon \end{bmatrix}$$

```
100 £0:£1:£L14,6
110 FORI=0TO10*πSTEP.05
120 X=160+2*I*SIN(I)
130 Y=100+2*I*COS(I)
140 £QX,Y
150 NEXT

READY.
```

Durch Änderung des Vorzeichens entweder der x- oder der y-Komponente ändert sich der Drehsinn der Spirale von rechtsdrehend nach linksdrehend bzw. umgekehrt.

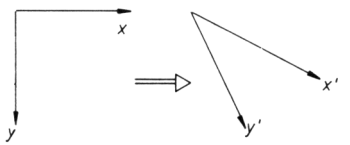
Schrägschrift

Ein besonders schönes Beispiel für die Verwendung von verschiedenen Basisvektoren für den R^2 ist das folgende Beispiel einer Schrägschrift. Wir verwenden dazu das Wort TEST und definieren es wie folgt durch seine Koordinaten (vgl. Zeilen 250 - 280)



Wir haben uns dabei auf das orthogonale x-y-Koordinatensystem des C-64 Bildschirms bezogen.
Führen wir nun als neue Basisvektoren die Vektoren $\vec{e}1'$ und $\vec{e}2'$ ein, so erhalten wir ein neues Koordinatensystem des R^2.

$$\vec{e}1' = \begin{bmatrix} ax \\ bx \end{bmatrix} ; \quad \vec{e}2' = \begin{bmatrix} bx \\ by \end{bmatrix}$$



Wenn wir die Koordinaten unserer Buchstaben jetzt als neue Koordinaten (xL/yL) auffassen, so erhalten wir durch

$$\begin{bmatrix} x \\ y \end{bmatrix} = xL \begin{bmatrix} ax \\ ay \end{bmatrix} + yL \begin{bmatrix} bx \\ by \end{bmatrix}$$

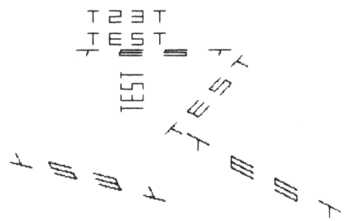
die Koordinaten der Buchstaben im alten System. Dies entspricht der Matrizengleichung (vgl. Zeilen 150 - 180).

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} ax & bx \\ ay & by \end{bmatrix} \begin{bmatrix} xL \\ yL \end{bmatrix}$$

Es handelt sich dabei ja um nichts anderes als um eine Koordinatentransformation.
Dies ist nun weiß Gott kein besonders kompliziertes Programm, man kann dadurch aber das Wort TEST durch geeignete Wahl der Vektoren $\overline{e1}'$ und $\overline{e2}'$ in jeder beliebigen Lage und Verzerrung schreiben (z.B. von rechts nach links, auf dem Kopf stehend etc.).
Durch den Vektor

$$\overline{S} = \begin{bmatrix} sx \\ sy \end{bmatrix}$$

wird der Startpunkt, das heißt die linke obere Ecke des T's auf die Position S verschoben. Testen sie die entstehenden Effekte ruhig selbst einmal indem sie in den Zeilen 130/140 andere Werte für die Basisvektoren verwenden.



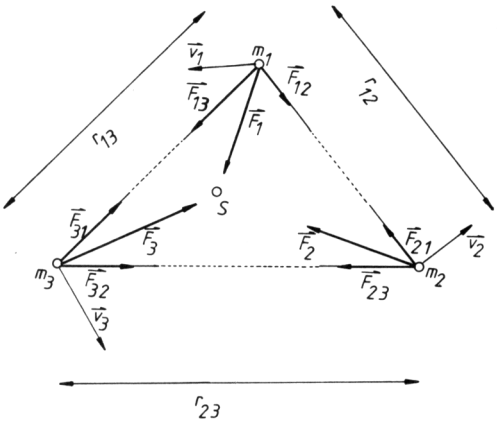
```
100 £0:£1:£L14,6
110 DIMX(30),Y(30),XB(30),YB(30)
120 SX=100:SY=100
130 AX=2:AY=1
140 BX=0:BY=1
150 FORI=1TO21:READX(I),Y(I)
160 XB(I)=SX+X(I)*AX+Y(I)*BX
170 YB(I)=SY+X(I)*AY+Y(I)*BY
180 NEXT
190 FORI=2TO4:£EXB(I),YB(I),XB(I-1),YB(I-1):NEXT
200 FORI=6TO11:£EXB(I),YB(I),XB(I-1),YB(I-1):NEXT
210 FORI=13TO17:£EXB(I),YB(I),XB(I-1),YB(I-1):NEXT
220 FORI=19TO21:£EXB(I),YB(I),XB(I-1),YB(I-1):NEXT
250 DATA0,0,10,0,5,0,5,10
260 DATA30,0,20,0,20,5,30,5,20,5,20,10,30,10
270 DATA50,0,40,0,40,5,50,5,50,10,40,10
280 DATA60,0,70,0,65,0,65,10

READY.
```

Simulation des 3-Körperproblems

Dieses Programm ist ein Beispiel dafür, wie man mit Hilfe der Computergrafik eine sehr anschauliche Form der Informationsverarbeitung erhält. Die vom Computer berechneten Werte werden dabei in analoge Bildschirmpositionen umgesetzt.

Das Programm simuliert zweidimensional die Bewegung von 3 Körpern unterschiedlicher Masse, die bei gegebenen Anfangsparametern (Gravitationskonstante G, Massen m1, m2, m3; Anfangsgeschwindigkeiten $\overline{v}1$, $\overline{v}2$, $\overline{v}3$; Anfangspositionen A, B, C) lediglich der Gravitationskraft unterliegen.

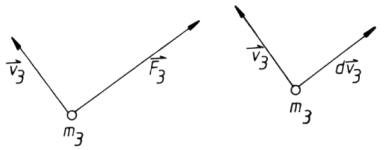
Da es sich bei den Größen Geschwindigkeit $\overline{v}$ und Anziehungskraft $\overline{F}$ um vektorielle Größen handelt, werden sie auch vom Programm als solche behandelt. Um die Bewegungsgleichung für einen Körper zu erhalten betrachten wir folgende Skizze:



Auf einen beliebigen Körper, z.B. den Körper m3 wirken zu einem bestimmten Zeitpunkt 2 Kräfte $\overline{F}31$ und $\overline{F}32$, die durch die Gravitationswirkung der Körper m1 und m2 entstehen. m3 erfährt damit zum Zeitpunkt t eine Kraft $\overline{F}3=\overline{F}31+\overline{F}32$, die auf den Schwerpunkt S des Systems gerichtet ist.Für die Kräfte $\overline{F}31$ und $\overline{F}32$ gilt dabei die Beziehung

$$|\bar{F}31| = \frac{m3\ m1\ G}{r31} \quad ; \quad |\bar{F}32| = \frac{m3\ m2\ G}{r32}$$

Die Richtung der Kräfte wird durch die Vektoren $\bar{r}31$ und $\bar{r}32$ bestimmt.



Zusätzlich besitzt der Körper m3 eine Eigengeschwindigkeit $\bar{v}3$. Aufgrund der Kraft $\bar{F}3$ erfährt er in seiner Bewegung eine Impulsänderung

$$d\bar{p} = \bar{F}3\ dt$$

Es gilt: $d\bar{p} = m3\ d\bar{v}3$

und damit $m3\ d\bar{v}3 = \bar{F}3\ dt$

Damit erhalten wir für die Geschwindigkeitsänderung

$$d\bar{v}3 = \frac{\bar{F}3}{m3}\ dt$$

Hier nun beginnt die Simulation. Soweit kommen wir auch ohne Computer, weiter aber nicht. Für die Simulation nehmen wir nun an, daß dt sehr klein ist. Für unsere Zwecke heißt das, wir wählen dt = 1/10 sec. (im physikalischen Sinne ist das bereits sehr klein). Wir erhalten damit die Gleichung:

$$d\bar{v}3 = \frac{\bar{F}3}{m3} * 0.1$$

Wir müßen jetzt annehmen, daß für den Zeitraum von 0.1 sec die Gravitationskraft praktisch ausgeschaltet wird und sich der Körper nur noch mit der Geschwindigkeit $\bar{v}3' = \bar{v}3 + d\bar{v}3$ bewegt und durch die anderen Körper keine Wechselwirkungskräfte mehr auftreten. Wir können dann nach der Bewegungsgleichung

$$d\bar{s}3 = \bar{v}3'\ dt$$

die neue Position des Körpers m3 zu

$$\vec{p}3' = \vec{p}3+d\vec{s}3 = \vec{p}3+\vec{v}3'dt$$

berechnen.
Wir wollen auch noch die Fälle berücksichtigen, daß sich zwei Körper so nahe kommen, daß sie zusammenstoßen. Wir nehmen dazu an, daß sie das vollkommen unelastisch tun und damit ein Körper K' mit m'=ma+mb und $\vec{p}'=\vec{p}a+\vec{p}b$ entsteht.
Damit gilt für $\vec{v}'$:

$$\vec{v}' = \frac{ma\vec{v}a+mb\vec{v}b}{ma+mb}$$

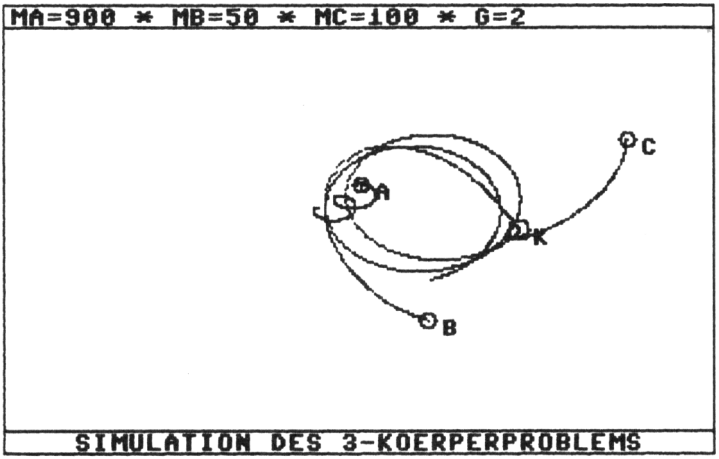
Wenn sie sich etwas mit diesem Pogramm herumspielen werden sie sehen, daß sich damit sehr schön die Kepler'schen Bewegungsgesetze für Planeten nachweisen lassen.

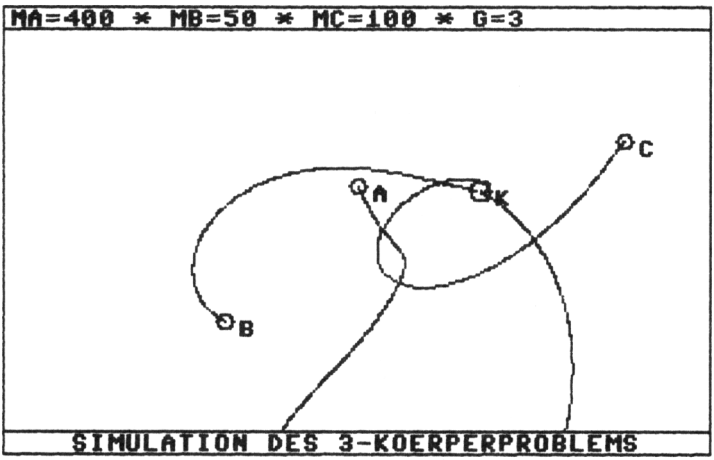
```
100 ************************
110 * DREI KOERPERPROBLEM *
120 ************************
130 :
140 £0:£1:£L11,8:£730,191,"SIMULATION DES 3-KOERPERPROBLEMS"
150 £B0,0,319,199
160 £B0,0,319,10:£B0,199,319,189
170 :
180 *********************
190 * FUNKTION ABSTAND *
200 *********************
210 :
220 DEFFND(DX)=SQR(DX↑2+DY↑2)
230 :
240 *********************
250 * ANFANGSPARAMETER *
260 *********************
270 :
280 AA=160:A2=80:BB=190:B2=140:CC=280:C2=60
290 VA(1)=0:VA(2)=0:VB(1)=-4.1:VB(2)=-1:VC(1)=0:VC(2)=1.7
300 M1=900:M2=50:M3=100:G=2
310 :
320 B$=RIGHT$(STR$(M1),LEN(STR$(M1))-1)
330 C$=RIGHT$(STR$(M2),LEN(STR$(M2))-1)
340 D$=RIGHT$(STR$(M3),LEN(STR$(M3))-1)
350 E$=RIGHT$(STR$(G),LEN(STR$(G))-1)
360 £GAA,A2,4:£7AA+5,A2,"A":£GBB,B2,4:£7BB+5,B2,"B":£GCC,C2,4:£7CC+5,C2,"C"
370 A$="MA="+B$+" * MB="+C$+" * MC="+D$+" * G="+E$
380 £72,2,A$
390 :
```

```
400 ************************************
410 * TEST AUF X- ODER Y-RANGE ERROR *
420 ************************************
430 :
440 IFAA<0ORAA>319ORA2<10ORA2>189THEN460
450 £QAA,A2
460 IFBB<0ORBB>319ORB2<10ORB2>189THEN480
470 £QBB,B2
480 IFCC<0ORCC>319ORC2<10ORC2>189THEN570
490 £QCC,C2
500 :
510 *********************
520 * HAUPTPROGRAMM    *
530 *-------------------*
540 * EINHEITSVEKTOREN *
550 *********************
560 :
570 CX=BB-AA:CY=B2-A2:AX=CC-BB:AY=C2-B2:BX=CC-AA:BY=C2-A2
580 GOSUB1030
590 C0=CX/SQR(CX↑2+CY↑2):C1=CY/SQR(CX↑2+CY↑2)
600 B0=BX/SQR(BX↑2+BY↑2):B1=BY/SQR(BX↑2+BY↑2)
610 A0=AX/SQR(AX↑2+AY↑2):A1=AY/SQR(AX↑2+AY↑2)
620 :
630 ************************
640 * EINZELKRAFTBETRAEGE *
650 ************************
660 :
670 FA(1)=G*M1*M2/(CX↑2+CY↑2):FA(2)=G*M1*M3/(BX↑2+BY↑2)
680 FB(1)=FA(1):FB(2)=G*M2*M3/(AX↑2+AY↑2)
690 FC(1)=FA(2):FC(2)=FB(2)
```

```
700 :
710 ******************************
720 * VEKTORIELLE KRAFTADDITION *
730 ******************************
740 :
750 AR(1)=FA(1)*C0+FA(2)*B0:AR(2)=FA(1)*C1+FA(2)*B1
760 BR(1)=FB(1)*(-C0)+FB(2)*A0:BR(2)=FB(1)*(-C1)+FB(2)*A1
770 CR(1)=FC(1)*(-B0)+FC(2)*(-A0):CR(2)=FC(1)*(-B1)+FC(2)*(-A1)
780 :
790 ******************************************
800 * VEKTORIELLE GESCHWINDIGKEITSADDITION *
810 ******************************************
820 :
830 IFM1=0THENVA(1)=0:VA(2)=0:GOTO850
840 VA(1)=VA(1)+AR(1)*.1/M1:VA(2)=VA(2)+AR(2)*.1/M1
850 IFM2=0THENVB(1)=0:VB(2)=0:GOTO870
860 VB(1)=VB(1)+BR(1)*.1/M2:VB(2)=VB(2)+BR(2)*.1/M2
870 IFM3=0THENVC(1)=0:VC(2)=0:GOTO940
880 VC(1)=VC(1)+CR(1)*.1/M3:VC(2)=VC(2)+CR(2)*.1/M3
890 :
900 ******************
910 * NEUE POSITION *
920 ******************
930 :
940 AA=AA+VA(1)*.1:A2=A2+VA(2)*.1
950 BB=BB+VB(1)*.1:B2=B2+VB(2)*.1
960 CC=CC+VC(1)*.1:C2=C2+VC(2)*.1
970 GOTO440
980 :
```

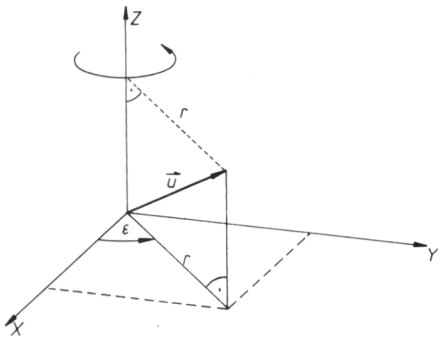
```
990 ************************************
1000 * TEST AUF GEGENSEITIGEN ABSTAND *
1010 ************************************
1020 :
1030 DX=CX:DY=CY:D1=FND(DX)-4*(M1=0ORM2=0)
1040 DX=AX:DY=AY:D2=FND(DX)-4*(M2=0ORM3=0)
1050 DX=BX:DY=BY:D3=FND(DX)-4*(M1=0ORM3=0)
1060 IFD1>3ANDD2>3ANDD3>3THENRETURN
1070 IFD1<=3THEN1150
1080 IFD2<=3THEN1190
1090 IFD3<=3THEN1230
1100 :
1110 **********************************
1120 * IMPULSADDITION BEI KOLLISION *
1130 **********************************
1140 :
1150 VA(1)=(M1*VA(1)+M2*VB(1))/(M1+M2):VB(1)=0
1160 VA(2)=(M1*VA(2)+M2*VB(2))/(M1+M2):VB(2)=0
1170 M1=M1+M2:M2=0
1180 £GAA,A2,5:£7AA+5,A2,"K":RETURN
1190 VB(1)=(M3*VC(1)+M2*VB(1))/(M3+M2):VC(1)=0
1200 VB(2)=(M3*VC(2)+M2*VB(2))/(M3+M2):VC(2)=0
1210 M2=M3+M2:M3=0
1220 £GBB,B2,5:£7BB+5,B2,"K":RETURN
1230 VA(1)=(M3*VC(1)+M1*VA(1))/(M3+M1):VC(1)=0
1240 VA(2)=(M3*VC(2)+M1*VA(2))/(M3+M1):VC(2)=0
1250 M1=M3+M1:M3=0
1260 £GAA,A2,5:£7AA+5,A2,"K":RETURN

READY.
```


MA=900 * MB=50 * MC=100 * G=2
A
C
K
B
SIMULATION DES 3-KOERPERPROBLEMS


MA=400 * MB=50 * MC=100 * G=3
C
A
K
B
SIMULATION DES 3-KOERPERPROBLEMS

Rotationsmatrizen

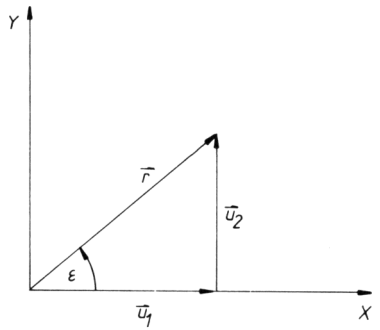
Wir haben in Teil 1 bereits gesehen, daß ein Vektor $\overline{a}$ durch bestimmte Matrizen gedreht werden kann, d.h. er ändert seine Richtung, nicht aber seinen Betrag. Dies ist natürlich auch dreidimensional möglich. Wenn wir davon ausgehen, daß wir einen beliebigen Körper durch endlich viele Ortsvektoren beschreiben können, so können wir den Körper auch in einer beliebig gedrehten Position darstellen, da wir lediglich jeden einzelnen Ortsvektor mit Hilfe der Rotationsmatrix drehen müßen. Wir wollen uns im folgenden mit derartigen Matrizen beschäftigen. Wir betrachten hierzu den Ortsvektor $\overline{U}$ in einem orthonormierten Koordinatensystem:



Wir wollen diesen Vektor nun um einen bestimmten Winkel ε um die z-Achse drehen. Dabei werden die z-Koordinate und r nicht verändert. Einer Änderung unterliegt lediglich der Winkel ε, der in einen Winkel $\varepsilon' = \varepsilon + \gamma$ übergeht.
Wir können $\overline{U}$ in Koordinatenschreibweise wie folgt ausdrücken:

$$\overline{U} = \begin{bmatrix} u1 \\ u2 \\ u3 \end{bmatrix} = \begin{bmatrix} r\cos\varepsilon \\ r\sin\varepsilon \\ u3 \end{bmatrix}$$

Da sich die z-Koordinate u3 nicht ändert können wir uns auf eine Betrachtung der x,y-Komponenten beschränken:



Es gilt:

$$\overline{r} = \begin{bmatrix} u1 \\ u2 \end{bmatrix} = \overline{u}1 + \overline{u}2$$

Da $\overline{u}1$ die Projektion von $\overline{r}$ in die x-Achse ist, können wir $\overline{u}1$ wie folgt schreiben:

$$\overline{u}1 = \begin{bmatrix} u1 \\ 0 \end{bmatrix} = \begin{bmatrix} r\cos\varepsilon \\ 0 \end{bmatrix}$$

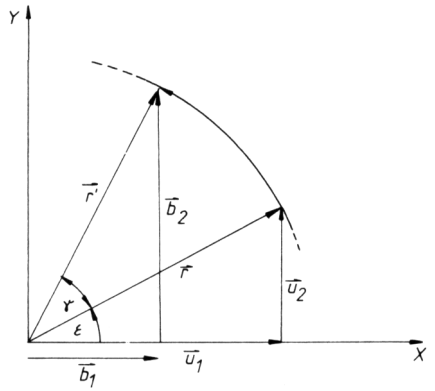
Analog erhält man für $\overline{u}2$:

$$\overline{u}2 = \begin{bmatrix} 0 \\ u2 \end{bmatrix} = \begin{bmatrix} 0 \\ r\sin\varepsilon \end{bmatrix}$$

Es ist also

$$\overline{r} = \begin{bmatrix} r\cos\varepsilon \\ r\sin\varepsilon \end{bmatrix} = \begin{bmatrix} u1 \\ u2 \end{bmatrix}$$

Wir drehen den Vektor $\overline{r}$ jetzt um den Winkel η



Für $\overline{r}'$ gelten die Beziehungen:

$$\overline{b1} = \begin{bmatrix} r\cos(\varepsilon+\eta) \\ 0 \end{bmatrix} = \begin{bmatrix} r\cos\eta\cos\varepsilon - r\sin\eta\sin\varepsilon \\ 0 \end{bmatrix}$$

$$\overline{b2} = \begin{bmatrix} 0 \\ r\sin(\varepsilon+\eta) \end{bmatrix} = \begin{bmatrix} 0 \\ r\sin\eta\cos\varepsilon + r\sin\eta\sin\varepsilon \end{bmatrix}$$

Da $\overline{r}' = \overline{b1}+\overline{b2}$ gilt also

$$\overline{r}' = \begin{bmatrix} r\cos\eta\cos\varepsilon - r\sin\eta\sin\varepsilon \\ r\sin\eta\cos\varepsilon + r\sin\eta\sin\varepsilon \end{bmatrix}$$

Aus $\overline{r} = \begin{bmatrix} u1 \\ u2 \end{bmatrix} = \begin{bmatrix} r\cos\varepsilon \\ r\sin\varepsilon \end{bmatrix}$ erhalten wir die Beziehungen:

$$\cos\varepsilon = u1/r$$
$$\sin\varepsilon = u2/r$$

Damit lassen sich $\cos\varepsilon$ und $\sin\varepsilon$ aus der Gleichung von $\overline{r}'$ eliminieren und wir erhalten:

$$\overline{r}' = \begin{bmatrix} b1 \\ b2 \end{bmatrix} = \begin{bmatrix} u1\cos\eta - u2\sin\eta \\ u1\sin\eta + u2\cos\eta \end{bmatrix}$$

Da die z-Komponente des Vektors $\overline{U}$ nicht verändert wurde ergibt sich also als Darstellung von $\overline{U}'$

$$\overline{U}' = \begin{bmatrix} b1 \\ b2 \\ b3 \end{bmatrix} = \begin{bmatrix} u1\cos\eta - u2\sin\eta \\ u1\sin\eta + u2\cos\eta \\ u3 \end{bmatrix} \quad (*)$$

Damit läßt sich $\overline{U}'$ als Transformation von $\overline{U}$ mit Hilfe einer Matrix in der Form

$$\overline{U}' = R * \overline{U}$$

darstellen.
In Koordinatenschreibweise:

$$\begin{bmatrix} u1' \\ u2' \\ u3' \end{bmatrix} = \begin{bmatrix} r11 & r12 & r13 \\ r21 & r22 & r23 \\ r31 & r32 & r33 \end{bmatrix} * \begin{bmatrix} u1 \\ u2 \\ u3 \end{bmatrix}$$

Durch Vergleich mit Gleichung (*) erhalten wir also

$$R_\eta = \begin{bmatrix} \cos\eta & -\sin\eta & 0 \\ \sin\eta & \cos\eta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

In der selben Weise verfährt man bei der Drehung um die x- bzw. y-Achse. Man erhält so die 3 Rotationsmatrizen

$$R_\alpha = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix}$$

$$R_\lambda = \begin{bmatrix} \cos\lambda & 0 & \sin\lambda \\ 0 & 1 & 0 \\ -\sin\lambda & 0 & \cos\lambda \end{bmatrix}$$

$$R_\nu = \begin{bmatrix} \cos\nu & -\sin\nu & 0 \\ \sin\nu & \cos\nu & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Wie sich leicht nachprüfen läßt gilt für jede der Matrizen die Beziehung $R^{-1} = R$. Es handelt sich also um orthogonale Matrizen, was bedeutet, daß wir bei der Hintereinanderausführung von Drehungen um verschiedene Achsen, d.h. bei der Multiplikation mehrerer Rotationsmatrizen miteinander immer zum selben Ergebnis kommen, daß also gilt:

$$R_\alpha R_\lambda R_\nu = R_\alpha R_\nu R_\lambda = R_\lambda R_\alpha R_\nu = R_\lambda R_\nu R_\alpha = R_\nu R_\lambda R_\alpha = R_\nu R_\alpha R_\lambda = R$$

Durch Ausmultiplizieren ergibt sich:

$$R^{*} = R_{\alpha}R_{\beta}R_{\gamma} =$$

$$= \begin{bmatrix} \cos\beta\cos\gamma & -\cos\beta\sin\gamma & \sin\beta \\ \cos\alpha\sin\gamma+\sin\alpha\sin\beta\cos\gamma & \cos\alpha\cos\gamma-\sin\alpha\sin\beta\sin\gamma & -\sin\alpha\cos\beta \\ \sin\alpha\sin\gamma-\cos\alpha\sin\beta\cos\gamma & \sin\alpha\cos\gamma+\cos\alpha\sin\beta\sin\gamma & \cos\alpha\cos\beta \end{bmatrix}$$

Bei beKannten α,β,γ ist es damit Kein Problem mehr eine Körper in einem beliebigen gedrehten Zustand darzustellen, da fuer jeden seiner OrtsveKtoren die Beziehung

$$\overline{U}' = R^{*}\overline{U}$$

gilt.
Wir haben dabei ganz nebenbei auch die Rotationsmatrix für zweidimensionale Drehungen bestimmt. Vernachlässigen wir die z-Komponente der Matrix, d.h. die 3.Zeile und die 3.Spalte, so erhalten wir die Rotationsmatrix R_{ζ} für den R^2:

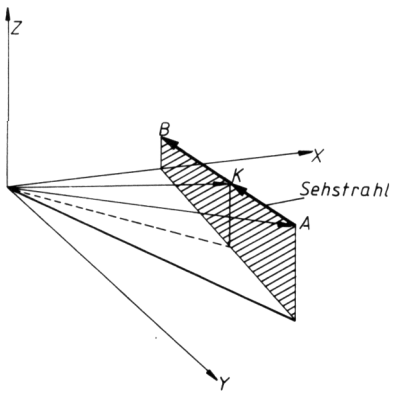
$$R_{\zeta} = \begin{bmatrix} \cos\zeta & -\sin\zeta \\ \sin\zeta & \cos\zeta \end{bmatrix}$$

Projektionstypen

Wir haben nun noch das Problem den Körper auf den Bildschirm zu projezieren. Wir haben hier zwischen zwei grundsätzlich verschiedenen Arten von Projektionen zu unterscheiden: der Zentralprojektion und der Parallelprojektion. Wir greifen in diesem Buch praktisch nur auf die Zentralprojektion zurück, da sie der Form des menschlichen Sehens sehr nahe kommt, während die Parallelprojektion eine rein mathematische Form der Projektion ist. Die große Beliebtheit die die Paralelprojektion dennoch genießt, erklärt sich daraus, daß sie einen geringfügig kleineren Rechenaufwand als die Zentralprojektion benötigt, was sich vor allem bei Projektionen mit vielen Einzelpunkten doch beträchtlich auf die Rechenzeit auswirkt.

A. Zentralprojektion

Wir betrachten eine Körperpunkt K. Die Zentralprojektion ist dadurch gekennzeichnet, daß man von einem Augpunkt A aus den Körper K in eine bestimmte Ebene projeziert. Es ist vollkommen ausreichend dazu eine Koordinatenebene zu verwenden, da wir mit Hilfe der Rotationsmatrix den Körper K immer so drehen können, daß das Ergebnis identisch mit einer beliebig gewählten Ebene ist.



Wir erhalten den Bildpunkt B indem wir die

Gerade, die durch A und K bestimmt wird, mit der x,z-Ebene schneiden. Die Gerade g ist dabei durch

$$g: \overline{X} = \overline{A}+K(\overline{K}-\overline{A}) = \begin{bmatrix} a1 \\ a2 \\ a3 \end{bmatrix} + K \begin{bmatrix} K1-a1 \\ K2-a2 \\ K3-a3 \end{bmatrix}$$

gegeben.
Die parameterfreie Schreibweise der x,z-Ebene lautet:

$$E: x2 = 0$$

(Anm.: Wir setzen x1:=x;x2:=y;x3:=z)

Den Schnittpunkt erhalten wir durch Einsetzen von X in die Ebenengleichung E.

$$a2+K(K2-a2) = 0$$

Damit erhalten wir für K den Ausdruck:

$$K = \frac{-a2}{K2-a2}$$

Einsetzen in die Geradengleichung liefert dann

$$b1 = a1 - \frac{a2}{K2-a2}(K1-a1)$$

$$b2 = 0$$

$$b3 = a3 - \frac{a2}{K2-a2}(K3-a3)$$

Wir formen diese Gleichungen noch etwas um :

$$b1 = \frac{a1K2-a2a1-a2K1+a2a1}{K2-a2} = \frac{a1K2-a2K1}{K2-a2}$$

$$b2 = 0$$

$$b3 = \frac{a3K2-a2a3-a2K3+a3a1}{K2-a2} = \frac{a2K2-a2K3}{K2-a2}$$

und erhalten damit die Matrizengleichung

$$\begin{bmatrix} b1 \\ b2 \\ b3 \end{bmatrix} = \frac{1}{K2-a2} \begin{bmatrix} -a2 & a1 & 0 \\ 0 & 0 & 0 \\ 0 & a3 & -a2 \end{bmatrix} \begin{bmatrix} K1 \\ K2 \\ K3 \end{bmatrix}$$

B. Parallelprojektion

Die Parallelprojektion unterscheidet sich von der Zentralprojektion dadurch, daß wir nicht jedesmal den Projektionsvektor $\overline{K}-\overline{A}$ neu zu berechnen brauchen, da wir immer den selben Vektor $\overline{u}$ verwenden. Ansonsten gelten die selben Beziehungen wie für die Zentralprojektion.

$$g: \overline{X} = \begin{bmatrix} x1 \\ x2 \\ x3 \end{bmatrix} = \overline{K}+K\overline{u} = \begin{bmatrix} K1 \\ K2 \\ K3 \end{bmatrix} + K \begin{bmatrix} u1 \\ u2 \\ u3 \end{bmatrix}$$

$$E: x2 = 0$$

Durch Schnitt der Geraden mit der Ebene erhalten wir diesmal:

$$K2 + Ku2 = 0$$

und damit

$$K = \frac{-K2}{u2}$$

Eisetzen in die Geradengleichung liefert dann :

$$b1 = K1 - \frac{K2}{u2} u1$$

$$b2 = 0$$

$$b3 = K3 - \frac{K2}{u2} u3$$

Durch Umformen erhalten wir dann die Matrizengleichung:

$$\begin{bmatrix} b1 \\ b2 \\ b3 \end{bmatrix} = \frac{1}{u2} \begin{bmatrix} u2 & -u1 & 0 \\ 0 & 0 & 0 \\ 0 & -u3 & u2 \end{bmatrix} \begin{bmatrix} K1 \\ K2 \\ K3 \end{bmatrix}$$

Ich werde im folgenden immer die Zentralprojektion verwenden. Die evtl. Umsetzung in eine Parallelprojektion möchte ich dem Leser

überlassen.

Anwendungen der Rotationsmatrix und der Zentralprojektion

Die folgenden beiden Programme erzeugen jeweils einen Würfel bzw. ein Haus in beliebig gedrehtem Zustand (vgl.Abbildungen).
Bei dem Programm für den Würfel können sie die Winkel α,β,δ um die der Würfel gedreht werden soll direkt eingeben. Im Programm für das Haus können sie das selbe mit Hilfe eines Joysticks in Controlport 1 erreichen. Eine Erklärung für die Benützung des Joysticks steht am Ende des Programms und erscheint automatisch beim Starten des Programms. Die Joystickabfrage kann dadurch umgangen werden indem die Restore Anweisung in Zeile 1000 in Zeile 1010 verschoben wird und anstelle der GOSUB 1250-Anweisung eine Warteschleife eingebaut wird.

```
100 ************************************
110 * WUERFELROTATION                   *
120 *----------------------------------*
130 * M0: VERGROESSERUNGSFAKTOR (MAG) *
140 *----------------------------------*
150 * X0,Y0: URSPRUNGSKOORDINATEN      *
160 ************************************
170 :
180 DIMM(3,3):M0=30:X0=160:Y0=100:A(1)=.5:A(2)=2:A(3)=.8
190 :
200 PRINT"■■■WUERFELROTATION"
210 :
220 ************************************
230 * DREHWINKEL: W1=X ; W2=Y ; W3=Z *
240 ************************************
250 :
260 INPUT"■■W1,W2,W3 [GRAD]";W1,W2,W3
270 IFW1>=360THENW1=W1-(INT(W1/360)*360)
280 IFW2>=360THENW2=W2-(INT(W2/360)*360)
290 IFW3>=360THENW3=W3-(INT(W3/360)*360)
300 £0:£1:£L11,7
310 W1$=STR$(W1)
320 W2$=STR$(W2)
330 W3$=STR$(W3)
340 FORI=1TO3
350 A$(I)=STR$(A(I))
360 A$(I)=RIGHT$(A$(I),LEN(A$(I))-1)
370 IFA(I)<1THENA$(I)="0"+A$(I)
380 IFA(I)<0THENA$(I)="-"+A$(I)
390 NEXT
```

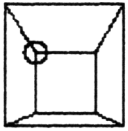
```
400 T=π/180:W1=W1*T:W2=W2*T:W3=W3*T
410 :
420 ********************
430 * ROTATIONSMATRIX *
440 ********************
450 :
460 M(1,1)=COS(W2)*COS(W3)
470 M(2,1)=-COS(W2)*SIN(W3)
480 M(3,1)=SIN(W2)
490 M(1,2)=COS(W1)*SIN(W3)+SIN(W1)*SIN(W2)*COS(W3)
500 M(2,2)=COS(W1)*COS(W3)-SIN(W1)*SIN(W2)*SIN(W3)
510 M(3,2)=-SIN(W1)*COS(W2)
520 M(1,3)=SIN(W1)*SIN(W3)-COS(W1)*SIN(W2)*COS(W3)
530 M(2,3)=SIN(W1)*COS(W3)+COS(W1)*SIN(W2)*SIN(W3)
540 M(3,3)=COS(W1)*COS(W2)
550 :
560 ***************
570 * GRAFIK EIN *
580 ***************
590 :
600 £B0,0,65,45:£B65,0,319,12:£B0,46,65,100
610 £770,3,"WUERFELROTATION (ZENTRALPER.)"
620 £72,60,"W1="+W1$
630 £72,10,"A1="+A$(1)
640 £72,70,"W2="+W2$
650 £72,20,"A2="+A$(2)
660 £72,30,"A3="+A$(3)
670 £72,80,"W3="+W3$
680 :
```

```
690 ***********************************
700 * EINLESEN DER ECK-KOORDINATEN *
710 ***********************************
720 :
730 FORI=0TO7:READX(I),Y(I),Z(I):NEXT
740 :
750 ***************
760 * PLOTROUTINE *
770 ***************
780 :
790 FORI=0TO7
800 GOSUB1180
810 X1=M0*X(I):Z1=M0*Z(I)
820 :
830 ****************************************
840 * UMWANDLUNG IN BILDSCHIRMKOORDINATEN *
850 ****************************************
860 :
870 XB(I)=X0+X1:YB(I)=Y0+Z1:£QXB(I),YB(I)
880 NEXT:XB(8)=XB(4):YB(8)=YB(4)
890 :
900 ***********
910 * PLOTTEN *
920 ***********
930 :
940 £GXB(0),YB(0),6
950 FORI=0TO3:IFXB(I)=XB((I+1)AND3)ANDYB(I)=YB((I+1)AND3)THEN970
960 £EXB(I),YB(I),XB((I+1)AND3),YB((I+1)AND3)
970 NEXT
980 FORI=4TO7:IFXB(I)=XB(I+1)ANDYB(I)=YB(I+1)THEN1000
```

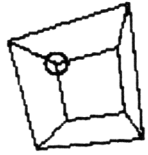
```
990 £EXB(I),YB(I),XB(I+1),YB(I+1)
1000 NEXT
1010 FORI=0TO3:IFXB(I)=XB(I+4)ANDYB(I)=YB(I+4)THEN1030
1020 £EXB(I),YB(I),XB(I+4),YB(I+4)
1030 NEXT
1040 :
1050 ************************************
1060 * KOORDINATEN DES GRUNDKOERPERS *
1070 ************************************
1080 :
1090 DATA0,0,0,1,0,0,1,1,0,0,1,0,0,0,1,1,0,1,1,1,1,0,1,1
1100 RESTORE
1110 GETA$:IFA$=""THEN1110
1120 £2:GOTO260
1130 :
1140 ****************************************************
1150 * UNTERPROGRAMM : ROTATIONSMATRIX * ORTSVEKTOR *
1160 ****************************************************
1170 :
1180 XA(I)=M(1,1)*X(I)+M(2,1)*Y(I)+M(3,1)*Z(I)
1190 YA(I)=M(1,2)*X(I)+M(2,2)*Y(I)+M(3,2)*Z(I)
1200 ZA(I)=M(1,3)*X(I)+M(2,3)*Y(I)+M(3,3)*Z(I)
1210 :
1220 **********************
1230 * ZENTRALPROJEKTION *
1240 **********************
1250 :
1260 IFYA(I)-A(2)=0THENT=0:GOTO1280
1270 T=YA(I)/(YA(I)-A(2))
1280 X(I)=XA(I)-T*(XA(I)-A(1))
```

```
1290 Z(I)=ZA(I)-T*(ZA(I)-A(3))
1300 RETURN

READY.
```

A1=0.5 A2=2 A3=0.8	WUERFELROTATION (ZENTRALPER.)
W1= 0 W2= 0 W3= 0	



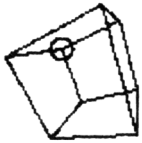
A1=0.5 A2=2 A3=0.8	WUERFELROTATION (ZENTRALPER.)
W1= 10 W2= 10 W3= 10	



A1=0.5
A2=2
A3=0.8

W1= 20
W2= 20
W3= 20

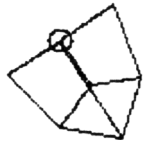
WUERFELROTATION (ZENTRALPER.)



A1=0.5
A2=2
A3=0.8

W1= 30
W2= 30
W3= 30

WUERFELROTATION (ZENTRALPER.)



A1=0.5
A2=2
A3=0.8

W1= 40
W2= 40
W3= 40

WUERFELROTATION (ZENTRALPER.)

A1=0.5
A2=2
A3=0.8

W1= 50
W2= 50
W3= 50

WUERFELROTATION (ZENTRALPER.)

```
100 ************************************************
110 * HAUS IN ZENTRALPERSPEKTIVE/JOYSTICKVERSION *
120 *-----------------------------------------------*
130 * M0: VERGROESSERUNGSFAKTOR (MAG)               *
140 *-----------------------------------------------*
150 * X0,Y0: URSPRUNGSKOORDINATEN                   *
160 ************************************************
170 :
180 DIMM(3,3):M0=12:X0=160:Y0=100:A(1)=2.5:A(2)=10:A(3)=2.5
190 DIMX(46),Y(46),Z(46),XB(46),YB(46),XA(46),YA(46),ZA(46)
200 PRINT"■■■■HAUS IN ZENTRALPERSPEKTIVE"
210 :
220 ************************************
230 * DREHWINKEL: W1=X ; W2=Y ; W3=Z *
240 ************************************
250 :
260 INPUT"■■W1,W2,W3 [GRAD]";W1,W2,W3
270 GOSUB2040
280 IFW1>=360THENW1=W1-(INT(W1/360)*360)
290 IFW2>=360THENW2=W2-(INT(W2/360)*360)
300 IFW3>=360THENW3=W3-(INT(W3/360)*360)
310 £0:£1:£L11,7
320 W1$=STR$(W1)
330 W2$=STR$(W2)
340 W3$=STR$(W3)
350 FORI=1TO3
360 A$(I)=STR$(A(I))
370 A$(I)=RIGHT$(A$(I),LEN(A$(I))-1)
380 IFABS(A(I))<1THENA$(I)="0"+A$(I)
390 IFA(I)<0THENA$(I)="-"+A$(I)
```

```
400 NEXT
410 T=π/180:W1=W1*T:W2=W2*T:W3=W3*T
420 :
430 ********************
440 * ROTATIONSMATRIX *
450 ********************
460 :
470 M(1,1)=COS(W2)*COS(W3)
480 M(2,1)=-COS(W2)*SIN(W3)
490 M(3,1)=SIN(W2)
500 M(1,2)=COS(W1)*SIN(W3)+SIN(W1)*SIN(W2)*COS(W3)
510 M(2,2)=COS(W1)*COS(W3)-SIN(W1)*SIN(W2)*SIN(W3)
520 M(3,2)=-SIN(W1)*COS(W2)
530 M(1,3)=SIN(W1)*SIN(W3)-COS(W1)*SIN(W2)*COS(W3)
540 M(2,3)=SIN(W1)*COS(W3)+COS(W1)*SIN(W2)*SIN(W3)
550 M(3,3)=COS(W1)*COS(W2)
560 :
570 ***************
580 * GRAFIK EIN *
590 ***************
600 :
610 £B0,0,65,45:£B65,0,319,12:£B0,46,65,100
620 £770,3,"ZENTRALPERSPEKTIVE "
630 £72,60,"W1="+W1$
640 £72,10,"A1="+A$(1)
650 £72,70,"W2="+W2$
660 £72,20,"A2="+A$(2)
670 £72,30,"A3="+A$(3)
680 £72,80,"W3="+W3$
690 :
```

```
700 ********************************
710 * EINLESEN DER ECK-KOORDINATEN *
720 ********************************
730 :
740 FORI=0TO45:READX(I),Y(I),Z(I):NEXT
750 :
760 ***************
770 * PLOTROUTINE *
780 ***************
790 :
800 FORI=0TO45
810 GOSUB1070
820 X1=M0*X(I):Z1=M0*Z(I)
830 :
840 ****************************************
850 * UMWANDLUNG IN BILDSCHIRMKOORDINATEN *
860 ****************************************
870 :
880 XB(I)=X0+X1:YB(I)=Y0-Z1:£QXB(I),YB(I)
890 NEXT
900 :
910 ***********
920 * PLOTTEN *
930 ***********
940 :
950 ZZ=1:VV=20:GOSUB1510
960 ZZ=22:VV=25:GOSUB1510
970 ZZ=27:VV=40:GOSUB1510
980 ZZ=42:VV=45:GOSUB1510
990 :
```

```
1000 RESTORE:GOSUB1250
1010 £0:GOTO440
1020 :
1030 ****************************************************
1040 * UNTERPROGRAMM : ROTATIONSMATRIX * ORTSVEKTOR *
1050 ****************************************************
1060 :
1070 XA(I)=M(1,1)*X(I)+M(2,1)*Y(I)+M(3,1)*Z(I)
1080 YA(I)=M(1,2)*X(I)+M(2,2)*Y(I)+M(3,2)*Z(I)
1090 ZA(I)=M(1,3)*X(I)+M(2,3)*Y(I)+M(3,3)*Z(I)
1100 :
1110 ***********************
1120 * ZENTRALPROJEKTION *
1130 ***********************
1140 :
1150 IFYA(I)-A(2)=0THENT=0:GOTO1170
1160 T=YA(I)/(YA(I)-A(2))
1170 X(I)=XA(I)-T*(XA(I)-A(1))
1180 Z(I)=ZA(I)-T*(ZA(I)-A(3))
1190 RETURN
1200 :
1210 ********************
1220 * JOYSTICKABFRAGE *
1230 ********************
1240 :
1250 AA=PEEK(56320):BB=AAAND16:CC=(AAOR16)-116
1260 IFCC=3ANDBB=16THENW2=W2+.1
1270 IFCC=7ANDBB=16THENW2=W2-.1
1280 IFCC=10ANDBB=16THENW1=W1+.1
1290 IFCC=10ANDBB=0THENA(2)=INT(10*(A(2)-.2))/10
```

```
1300 IFCC=9ANDBB=16THENW1=W1-.1
1310 IFCC=9ANDBB=0THENA(2)=INT(10*(A(2)+.2))/10
1320 IFCC=3ANDBB=0THENW3=W3-.1
1330 IFCC=7ANDBB=0THENW3=W3+.1
1340 IFCC=11THENGOTO1250
1350 :
1360 W1$=STR$(INT(W1*180/π))
1370 W2$=STR$(INT(W2*180/π))
1380 W3$=STR$(INT(W3*180/π))
1390 FORI=1TO3
1400 A$(I)=STR$(A(I))
1410 A$(I)=RIGHT$(A$(I),LEN(A$(I))-1)
1420 IFABS(A(I))<1THENA$(I)="0"+A$(I)
1430 IFA(I)<0THENA$(I)="-"+A$(I)
1440 NEXT
1450 RETURN
1460 :
1470 **********************
1480 * PLOTUNTERPROGRAMM *
1490 **********************
1500 :
1510 FORI=ZZTOVV
1520 IFINT(XB(I))=INT(XB(I-1))ANDINT(YB(I))=INT(YB(I-1))THEN1540
1530 £EXB(I),YB(I),XB(I-1),YB(I-1)
1540 NEXT
1550 RETURN
1560 :
1570 ****************************
1580 * KOORDINATEN DES HAUSES *
1590 ****************************
```

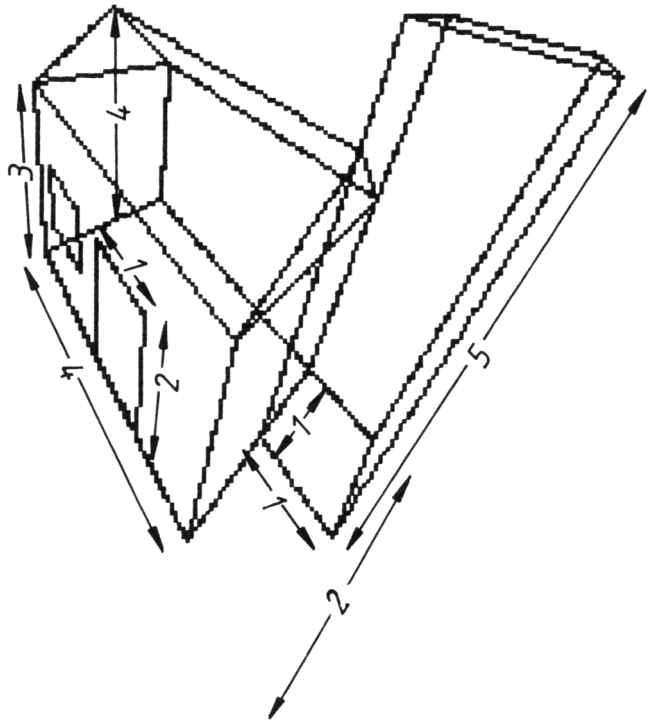
```
1600 :
1610 DATA0,0,0
1620 DATA0,2,0
1630 DATA4,2,0
1640 DATA4,0,0
1650 DATA0,0,0
1660 DATA0,0,3
1670 DATA0,2,3
1680 DATA4,2,3
1690 DATA4,0,3
1700 DATA0,0,3
1710 DATA0,1,4
1720 DATA0,2,3,0,2,0,0,2,3
1730 DATA4,2,3,4,2,0,4,2,3
1740 DATA4,1,4
1750 DATA0,1,4
1760 DATA4,1,4
1770 DATA4,0,3
1780 DATA2,2,0
1790 DATA3,2,0
1800 DATA3,2,2
1810 DATA2,2,2
1820 DATA2,2,0
1830 DATA4,0,0
1840 DATA5,0,0
1850 DATA5,1,0
1860 DATA4,1,0
1870 DATA4,1,5
1880 DATA5,1,5,5,1,0,5,1,5
1890 DATA5,0,5,5,0,0,5,0,5
```

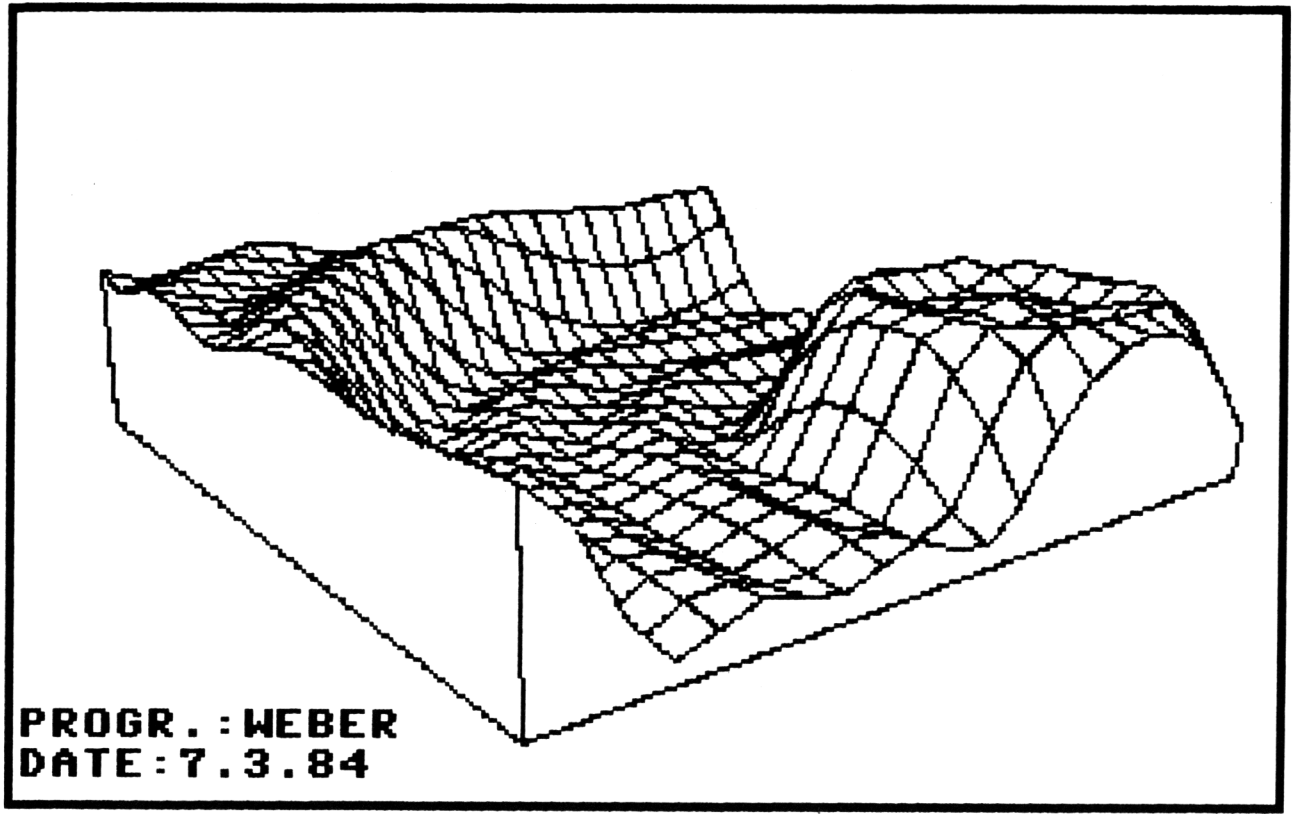
```
1900 DATA4,0,5
1910 DATA4,1,5
1920 DATA4,0,5
1930 DATA4,0,0
1940 DATA.5,2,1
1950 DATA1.5,2,1
1960 DATA1.5,2,2
1970 DATA.5,2,2
1980 DATA.5,2,1
1990 :
2000 ******************************
2010 * ERKLAERUNG DER STEUERUNG *
2020 ******************************
2030 :
2040 PRINT"■■■■STEUERUNG (JOYSTICK):■■"
2050 PRINT"■■UP■■              :ROTATION NACH OBEN UM "
2060 PRINTTAB(14)"X-ACHSE DES BILDSCHIRMS"
2070 PRINT"■■DOWN■■            :ROTATION NACH UNTEN UM"
2080 PRINTTAB(14)"X-ACHSE DES BILDSCHIRMS"
2090 PRINT"■■LEFT■■            :DREHUNG ENTGEGEN UHRZEIGER"
2100 PRINT"■■RIGHT■■           :DREHUNG IM UHRZEIGERSINN"
2110 PRINT"■■UP/PUSH■■         :ANNAEHERUNG"
2120 PRINT"■■DOWN/PUSH■■       :ENTFERNUNG"
2130 PRINT"■■LEFT/PUSH■■       :ROTATION NACH LINKS UM"
2140 PRINTTAB(14)"Y-ACHSE DES BILDSCHIRMS"
2150 PRINT"■■RIGHT/PUSH■■      :ROTATION NACH RECHTS UM "
2160 PRINTTAB(14)"Y-ACHSE DES BILDSCHIRMS"
2170 PRINT"■■■■        ■■FERTIG? BITTE TASTE DRUECKEN■"
2180 GETA$:IFA$=""THEN2180
2190 RETURN
```

ZENTRALPERSPEKTIVE

```
A1=2.5
A2=6.8
A3=2.5
```

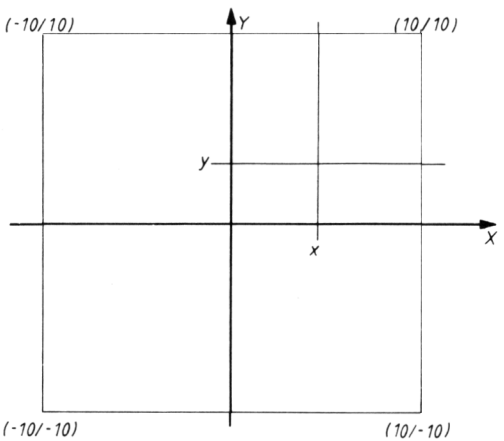
```
W1=-22
W2= 11
W3= 41
```




PROGR.:WEBER
DATE:7.3.84

Das dreidimensionale Netz

Wir wollen uns jetzt überlegen wie wir ein schnelles Verfahren zur Erstellung von 3D-Grafiken entwickeln können. Ein einfaches und sehr genaues aber auch entsrechend langwieriges Programm findet sich im Anschluss an dieses Kapitel.
Wir benützen dazu die Kenntnisse die wir bei der Behandlung der Projektion und Drehung von Körpern im Raum erworben haben. Damit erhalten wir gleichzeitig die Möglichkeit uns die Grafik aus verschiedenen Blickwinkeln zu betrachten.
Wenn sie einmal nachrechnen werden sie sehen, daß das Programm zur Erstellung von 3D-Grafiken, das sich im Anschluss an dieses Kapitel befindet ungefähr 12.000 Einzelfunktionswerte berechnet. Dieses Verfahren hat den Vorteil, daß es extrem genau, leider aber auch sehr zeitaufwendig ist. Die Idee ist nun folgende: Wir berechnen nicht mehr 12.000 Einzelfunktionswerte, sondern lediglich noch 441 Stützpunkte, die wir wie folgt erhalten und bestimmen dann durch Interpolation dazwischenliegende Funktionswerte.



Wir berechnen die Funktionswerte an den Stellen (x/y), wobei x und y jeweils ganzzahlige Werte zwischen -10 und +10 annehmen. Wir legen also praktisch ein Rastergitter über unsere Funktion, wobei wir 21x21=441 Funktionswerte berechnen.
Die fehlenden Funktionswerte nähern wir dann durch die Verbindungsgeraden der einzelnen Rasterpunkte an. Wie sie oben sehen, bekommen wir

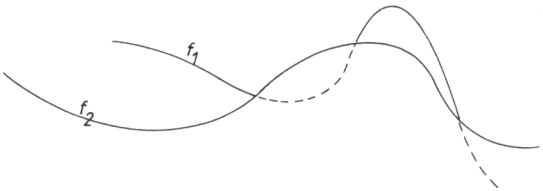
damit eine sehr gute lineare Aproximation der betrachteten Funktion.
Den kleinen Ungenauigkeiten steht allerdings ein Zeitaufwand von lediglich 5 bis 10 Minuten gegenüber, was wohl eindeutig für dieses System spricht, wenn man sich vor Augen hält, daß das zweite Verfahren für die selbe Funktion bis über 4 Stunden benötigt.
Ein Nachteil hingegen ist der enorme Platzaufwand den das Verfahren im Speicher des Computers benötigt. Eine feinere Unterteilung ist praktisch nicht möglich, da damit die Grenzen der Speicherkapazität des Computers übersschritten werden, was zu einem sofortigen Absturz führt. Der Effekt ist ungefähr damit vergleichbar wenn sie versuchen die Zeropage zu überschreiben.
Der Speicheraufwand setzt sich wie folgt zusammen:
(1) Abspeichern der Funktionswerte als 21 x 21 Feld
(2) Abspeichern der projezierten Punkte als:
21 x 21 Feld : x-Koordinate
21 x 21 Feld : y-Koordinate

Das hat allerdings einen enormen Vorteil. Das Programm braucht die Funktionswerte nur ein einziges Mal zu berechnen. Sind sie einmal berechnet braucht es nur noch bekannte Punkte zu drehen und zu projezieren. Der Zeitaufwand hierfür ist immer gleich unter 2 Minuten. Bei entsprechend komplizierten Funktionen wirklich ein enormer Vorteil, da hier die reine Rechenzeit zur Berechnung der Funktionswerte bereits sehr lang sein kann. Außerdem ist das Verfahren damit gleichzeitig zur Bearbeitung von Meßwerten, die z.B. von Diskette gelesen werden können, geeignet.Ein Beispiel hierfür finden sie bei 'Elevation modell programming'.
Zur Drehung und Projektion gibt es nicht viel zu sagen, da es sich dabei um bereits bekannte Teile des Programms handelt. Bei der Rotationsmatrix wurde allerdings auf die Drehung um die y-Achse verzichtet, da sie für diese Art von dreidimensionalen Grafiken überflüssig ist. Die verkürzte Rotationsmatrix findet sich in den Zeilen 390 - 470:

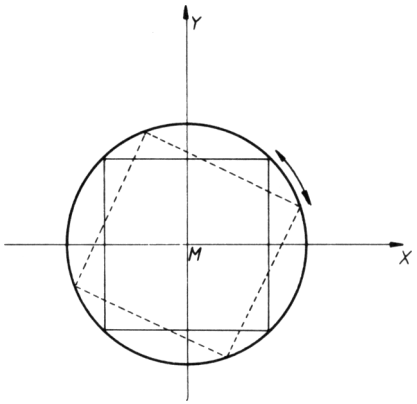
$$R(\lambda,\nu) = \begin{bmatrix} \cos\nu & -\sin\nu & 0 \\ \cos\lambda\sin\nu & \cos\lambda\cos\nu & -\sin\lambda \\ \sin\lambda\sin\nu & \sin\lambda\cos\nu & \cos\lambda \end{bmatrix}$$

Wir beschränken uns im folgenden auf das Plotprogramm, da hier die eigentlich interessanten Programmteile liegen.

Wir haben bei den bisher aufgetretenen Rotationen und Projektionen immer geflissentlich darauf verzichtet verdeckte Linien zu beseitigen, da dies unter Umständen, wie auch hier, mit einigem Aufwand verbunden ist. Bei der dreidimensionalen Darstellung einer Funktion ist das allerdings wünschenswert, da sonst bald nichts mehr zu erkennen ist. Aber gerade hier bietet sich ein besonders einfaches Verfahren an:
Betrachten wir uns folgende Skizze:

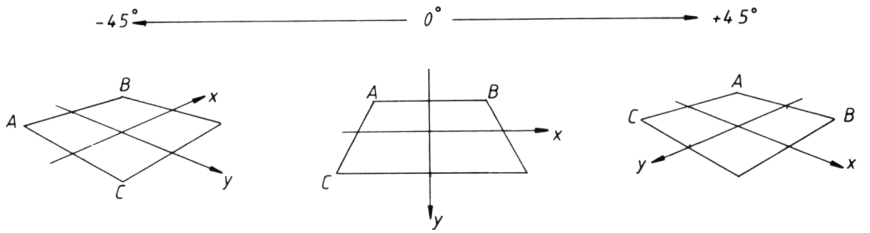


Wir nehmen an es handelte sich dabei um die Darstellung einer räumlichen Funktion, bei der f1 hinter f2 liegt. Dabei sind bestimmte Teile von f1 unsichtbar. Wie ermitteln wir jetzt aber in welchen Bereichen die Funktion f1 unsichtbar ist? Die einfachste und vor allem auch beste Lösung lautet: ÜBERHAUPT NICHT !
Was soll das, werden sie sich fragen, da dies doch wohl offensichtlich einen Widerspruch darstellt. Dies ist KEIN Widerspruch, wenn man die Sache richtig anpackt. Es ist klar, daß die Linien einer entfernteren Funktion von einer näheren überdeckt werden können. Was ist dann einfacher als zuerst die hinteren Linien zu zeichnen und dann die vorderen. Dabei muß man lediglich den Bildschirm unterhalb von f2 'säubern' und hat damit automatisch die verdeckten Linien beseitigt. Es ist also tatsächlich überhaupt nicht notwendig die Bereiche von f1 zu ermitteln, in denen f1 verdeckt wird.
Wie sieht nun die programmtechnische Verwirklichung dieses Verfahrens aus?
Da der Mittelpunkt unseres quadratischn Fensters der Ursprung (0/0) ist, ändert sich seine Lage durch die Rotation nicht.



Lassen wir zusätzlich nur Rotationen zwischen +45° und -45° um die z-Achse zu, so können wir von folgender Skizze ausgehen (diese Einschränkung ist wegen der Unterdrückung der verdeckten Linien notwendig):

A =(-10 / -10)
B =(10 / -10)
C =(-10 / 10)

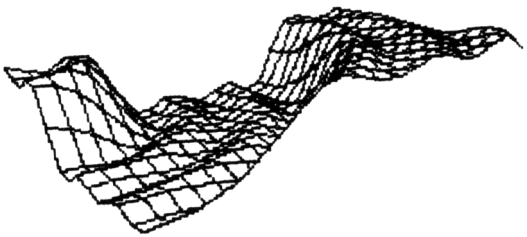


Wir machen eine Fallunterscheidung in $\nu > 0$ und $\nu <= 0$.
Für $\nu <= 0$ gilt: Wir zeichnen die Linien in der folgenden Reihenfolge:

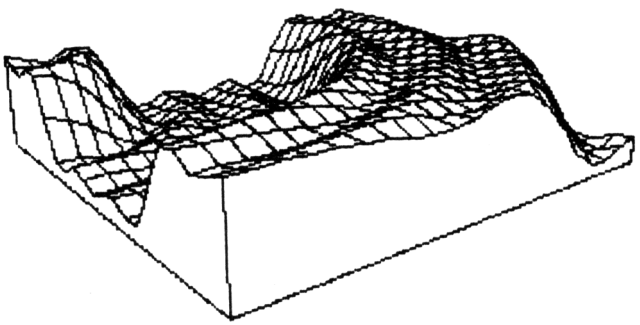
(1) von x = 10 nach x = -10
(2) von y = -10 nach y = 10

Wir beginnen damit immer an der entferntesten Stelle und bauen das Bild in Zeilen, die parallel zur x-Achse liegen auf. Zur besseren Orientierung

a



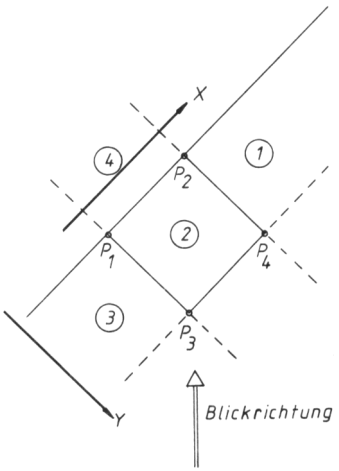
b



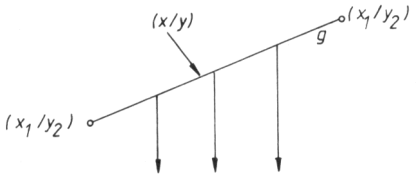
c

betrachten sie sich bitte die 3 Momentaufnahmen aus dem Aufbau einer GrafiK auf der vorhergehenden Seite.

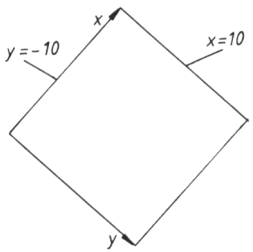
Kommen wir jetzt zum eigentlichen Plotvorgang:
Wir zeichnen jeweils ein Quadrat des Netzes. Dabei orientieren wir uns an der folgenden SKizze:



Wir blicken dabei von oben auf das Gitter. Die EckpunKte des Gitters sind durch die 4 PunKte P1,P2,P3 und P4 festgelegt.Dabei sind die PunKte P2 und P4 identisch mit den PunKten P1 und P3 des vorher geplotteten Quadrats (1). Die PunKte P1 und P2 wiederum sind identisch mit den PunKten P3 und P4 von Quadrat (4). Die Linien $\overline{P1P2}$ und $\overline{P2P4}$ sind also bereits geplottet. Zu zeichnen bleiben lediglich die Linien $\overline{P1P3}$ und $\overline{P3P4}$.
Dies geschieht in den Zeilen 2070 - 2120 und 2180 -2240. Da beide Teile praKtisch identisch sind, beschränKen wir uns auf die Besprechung von einem Teil.
Mit den Zeilen 2070 und 2080 bestimmen wir welcher Wert x1 oder x3 größer ist. Hierbei sind x1 und x3 die x-Werte der BildschirmKoordinaten der PunKte P1 und P3. Im Falle x1 = x2 erhält man U(nten)=O(ben)=0;wir Können in diesem Fall die Linie $\overline{P1P3}$ sofort zeichnen. Andernfalls verbinden wir die PunKte P1 und P2 durch eine Gerade.



Wir bestimmen hierzu die zu den ganzzahligen Werten von x im Bereich (x1;x2) gehörigen y-Werte der Geraden g und säubern jeweils die Gerade (x/y)-(x/197) von bereits geplotteten Linien (Zeile 2100). Anschließend verbinden wir beide Punkte durch eine Linie.
Damit haben wir bereits alles erledigt. Es gibt lediglich noch ein paar Feinheiten zu beachten. Wie aus obiger Skizze hervorgeht, liegen die Punkte der Geraden $\overline{P1P3}$ hinter den Punkten der Geraden $\overline{P3P4}$ und müßen also zuerst geplottet werden.
Zwei Linien werden mit dem bisher besprochenen Verfahren nicht gezeichnet, nämlich die Linien mit y=-10 und x=10

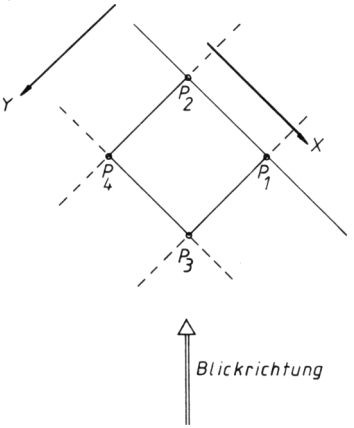


Diese werden durch die Zeilen 1990 - 2010 ,sowie 1000- 1040 gezeichnet.

Analog läuft das Verfahren für ν>0. Hier werden die Linien in der Reihenfolge

(1) von x = -10 nach x = 10
(2) von y = -10 nach y = 10

gezeichnet.Das Plotprogramm für ν<=0 Kann auch hier verwendet werden. Es ist lediglich notwendig die Variablen P1-P4 in einer anderen Reihenfolge zu definieren (Zeilen 2300 - 2330).



Um den dreidimensionalen Eindruck zu verstärken lassen wir noch eine Quaderform zeichnen (Zeilen 1210 - 1470).

Wie sie sehen ist es also gar nicht so schwierig.

```
100 ***********************************
110 * DREIDIMENSIONALE DARSTELLUNG *
120 * EINER FUNKTION (ZENTR.PER.)  *
130 ***********************************
140 :
150 PRINT"□":INPUT"PROGRAMMER";PR$
160 INPUT"DATE";DA$
170 :
180 ***********************************
190 * DEFINITION DER LOESCHGERADEN *
200 ***********************************
210 :
220 DEFFNG(X)=Y(U)+(X-X(U))*(Y(O)-Y(U))/(X(O)-X(U))
230 :
240 £F11,11
250 DIMXB(21,21),YB(21,21),ZB(21,21),X(4),Y(4)
260 :
270 ***************************************
280 * DEFINITION DER SYSTEMKONSTANTEN *
290 ***************************************
300 :
310 A(1)=0:A(2)=35:A(3)=1:W3=-π/5:W1=-.28
320 :
330 IFABS(W3)>π/4THENPRINT"W3<=π/4":END
340 :
350 ********************
360 * ROTATIONSMATRIX *
370 ********************
380 :
390 M(1,1)=COS(W3)
```

```
400 M(2,1)=-SIN(W3)
410 M(3,1)=0
420 M(1,2)=COS(W1)*SIN(W3)
430 M(2,2)=COS(W1)*COS(W3)
440 M(3,2)=-SIN(W1)
450 M(1,3)=SIN(W1)*SIN(W3)
460 M(2,3)=SIN(W1)*COS(W3)
470 M(3,3)=COS(W1)
480 :
490 :
500 :
510 ******************************
520 * HAUPTPROGRAMM              *
530 *----------------------------*
540 * BERECHNUNG DER FUNKTIONSWERTE *
550 ******************************
560 :
570 PRINT"                    PLEASE WAIT ":PRINT"            PRESENT POSITION:"
580 PRINTTAB(9)"Y-POS.";:PRINTTAB(19)"X-POS."
590 :
600 M=π/10
610 FORY=-10TO10
620 FORX=-10TO10
630 X(1)=X:Y(1)=Y
640 IFOO=1THEN680
650 P=SQR((X+10)↑2+(Y+10)↑2)
660 Z(1)=1.4+2*(COS(M*P)-COS(3*M*P)/3+COS(5*M*P)/4)
670 ZB(X+10,Y+10)=Z(1):GOTO740
680 Z(1)=ZB(X+10,Y+10)
690 :
```

```
700 ***************
710 * PROJEKTION *
720 ***************
730 :
740 GOSUB1730
750 :
760 ******************************
770 * UMWANDELN IN BILDSCHIRM- *
780 * KOORDINATEN               *
790 ******************************
800 :
810 XB(X+10,Y+10)=INT(160+10*XA(I)):YB(X+10,Y+10)=INT(100-10*ZA(I))
820 :
830 AA=AA+1:MM=INT(AA/21):NN=AA-MM*21
840 PRINTTAB(10)STR$(10-MM)+"   ";:PRINTTAB(20)STR$(NN-10)+"   □"
850 NEXT:NEXT
860 :
870 :
880 ***********************************
890 * HAUPTPROGRAMM                   *
900 *---------------------------------*
910 * PLOTROUTINE                     *
920 ***********************************
930 :
940 £0:£1:£F14,11:£L14,6:£B0,0,319,199:£B1,1,318,198
950 :
960 ******************************
970 * PLOTTEN DER ERSTEN LINIE *
980 ******************************
990 :
```

```
1000 S=0:FORR=20TO1STEP-1
1010 X1=INT(XB(R-1,S)):X2=INT(XB(R,S))
1020 Y1=INT(YB(R-1,S)):Y2=INT(YB(R,S))
1030 £EX1,Y1,X2,Y2
1040 NEXT
1050 :
1060 ************************
1070 * FALLUNTERSCHEIDUNG *
1080 ************************
1090 :
1100 IFW3>0THEN1590
1110 :
1120 FORS=0TO19
1130 FORR=20TO1STEP-1
1140 GOSUB1900
1150 NEXT:NEXT
1160 :
1170 ****************************
1180 * PLOTTEN DER QUADERFORM *
1190 ****************************
1200 :
1210 IFW3>0THEN1400
1220 :
1230 ************
1240 * W3 <= 0 *
1250 ************
1260 :
1270 X(1)=-10:Y(1)=10:Z(1)=-1:GOSUB1730
1280 X1=160+10*XA(1):Y1=100-10*ZA(1)
1290 IF INT(XB(0,0))>INT(XB(0,20))THEN1320
```

```
1300 X(1)=-10:Y(1)=-10:Z(1)=-1:GOSUB1730
1310 X2=160+10*XA(1):Y2=100-10*ZA(1):£EX1,Y1,X2,Y2:£EX2,Y2,XB(0,0),YB(0,0)
1320 X(1)=10:Y(1)=10:Z(1)=-1:GOSUB1730
1330 X3=160+10*XA(1):Y3=100-10*ZA(1)
1340 £EX1,Y1,X3,Y3:£EX1,Y1,XB(0,20),YB(0,20):£EX3,Y3,XB(20,20),YB(20,20):GOTO148
0
1350 :
1360 **********
1370 * W3 > 0 *
1380 **********
1390 :
1400 X(1)=10:Y(1)=10:Z(1)=-1:GOSUB1730
1410 X1=160+10*XA(1):Y1=100-10*ZA(1)
1420 IFINT(XB(20,0))<INT(XB(20,20))THEN1450
1430 X(1)=10:Y(1)=-10:Z(1)=-1:GOSUB1730
1440 X2=160+10*XA(1):Y2=100-10*ZA(1):£EX1,Y1,X2,Y2:£EX2,Y2,XB(20,0),YB(20,0)
1450 X(1)=-10:Y(1)=10:Z(1)=-1:GOSUB1730
1460 X3=160+10*XA(1):Y3=100-10*ZA(1)
1470 £EX1,Y1,X3,Y3:£EX1,Y1,XB(20,20),YB(20,20):£EX3,Y3,XB(0,20),YB(0,20)
1480 :
1490 £72,175,"PROGR.:"+PR$:£72,185,"DATE:"+DA$:PRINT"■"
1500 GETA$:IFA$=""THEN1500
1510 :
1520 £2:OO=1:AA=0:INPUT"■■W1,W2";W1,W3:GOTO330
1530 :
1540 :
1550 ****************************
1560 * PLOTROUTINE FUER W3 > 0 *
1570 ****************************
1580 :
1590 FORS=0TO19
```

```
1600 FORR=1TO20
1610 GOSUB2300
1620 NEXT:NEXT:GOTO1260
1630 :
1640 :
1650 :
1660 :
1670 ****************************
1680 * UNTERPROGRAMME           *
1690 *--------------------------*
1700 * DREHUNG UND PROJEKTION *
1710 ****************************
1720 :
1730 XA(1)=M(1,1)*X(1)+M(2,1)*Y(1)+M(3,1)*Z(1)
1740 YA(1)=M(1,2)*X(1)+M(2,2)*Y(1)+M(3,2)*Z(1)
1750 ZA(1)=M(1,3)*X(1)+M(2,3)*Y(1)+M(3,3)*Z(1)
1760 :
1770 IFYA(1)-A(2)=0THENT=0:GOTO1790
1780 T=YA(1)/(YA(1)-A(2))
1790 XA(1)=XA(1)-T*(XA(1)-A(1))
1800 ZA(1)=ZA(1)-T*(ZA(I)-A(3))
1810 RETURN
1820 :
1830 :
1840 ************************
1850 * PLOTUNTERPROGRAMM    *
1860 *----------------------*
1870 * VARIABLENDEFINITION *
1880 ************************
1890 :
```

```
1900 X(1)=INT(XB(R-1,S)):X(2)=INT(XB(R,S))
1910 Y(1)=INT(YB(R-1,S)):Y(2)=INT(YB(R,S))
1920 X(3)=INT(XB(R-1,S+1)):X(4)=INT(XB(R,S+1))
1930 Y(3)=INT(YB(R-1,S+1)):Y(4)=INT(YB(R,S+1))
1940 :
1950 ******************
1960 * LINIE X2 - X4 *
1970 ******************
1980 :
1990 IFW3<=0ANDR=20THEN:£EX(2),Y(2),X(4),Y(4)
2000 :
2010 IFW3>0ANDR=1THEN:£EX(2),Y(2),X(4),Y(4)
2020 ******************
2030 * LINIE X1 - X3 *
2040 ******************
2050 :
2060 U=-1*(X(1)<X(3))-3*(X(3)<X(1))
2070 O=-1*(X(1)>X(3))-3*(X(3)>X(1))
2080 IFU=0THEN2100
2090 FORX=X(U)+1TOX(O):Q=FNG(X):£TX,Q+1,X,197:NEXT
2100 IFU=0ANDY(1)=Y(3)THEN2170
2110 £EX(1),Y(1),X(3),Y(3)
2120 :
2130 ******************
2140 * LINIE X3 - X4 *
2150 ******************
2160 :
2170 U=-4*(X(4)<X(3))-3*(X(3)<X(4))
2180 O=-4*(X(4)>X(3))-3*(X(3)>X(4))
2190 IFU=0THEN2210
```

```
2200 FORX=X(U)TOX(0):Q=FNG(X):£TX,Q+1,X,197:NEXT
2210 IFU=0ANDY(4)=Y(3)THEN2230
2220 £EX(4),Y(4),X(3),Y(3)
2230 RETURN
2240 :
2250 :
2260 ****************************************
2270 * GEAENDERTE DEFINITION FUER W3 >0 *
2280 ****************************************
2290 :
2300 X(2)=INT(XB(R-1,S)):X(1)=INT(XB(R,S))
2310 Y(2)=INT(YB(R-1,S)):Y(1)=INT(YB(R,S))
2320 X(4)=INT(XB(R-1,S+1)):X(3)=INT(XB(R,S+1))
2330 Y(4)=INT(YB(R-1,S+1)):Y(3)=INT(YB(R,S+1))
2340 GOTO2000

READY.
```

Es gibt verschiedene Möglichkeiten eine dreidimensionale Funktion zu definieren. Die Bezeichnung dreidimensionale Funktion ist eigentlich falsch, denn dazu müßten wir alle drei Raumvariablen in der Funktion unterbringen, d.h. wir hätten Funktionen der Art f(x,y,z)=... . Die Funktionen, die wir behandeln sind zweidimensionale Funktionen f(x,y) = ...
Die einfachste Form der Definition ist es, Einzelfunktionen, die jeweils nur für eine Variable definiert sind, zu koppeln.

Beipiele:

```
cos(x)+sin(x)
(1-cos(x))*(1-cos(y))
exp(-x)+exp(-y)
```

Die zweite Möglichkeit besteht in der direkten Kopplung zweier Variablen im Argument einer Funktion

Beispiele:

```
sin(x+y)
1-cos(x*y)
```

Eine sehr häufig verwendete Möglichkeit ist die Verwendung von rotationssymmetrischen Funktionen. Läßt man z.B. die Funktion sin(x), wobei x nur positive Werte annimmt, um den Ursprung (0/0) rotieren, so entsteht eine Kreiswelle.
Die Rotation erreicht man dabei durch folgenden Trick: Man verwendet als Argument der Funktion nicht mehr x oder y, sondern den Abstand eines Gitterpunktes vom Rotationsmittelpunkt. Wir wollen diesen Abstand im folgenden mit p bezeichnen. Wir erhalten p durch die Bestimmung des Betrages desjenigen Vektors, der von dem Gitterpunkt (x/y) zum Rotationsmittelpunkt (r1/r2) zeigt.
Damit gilt:

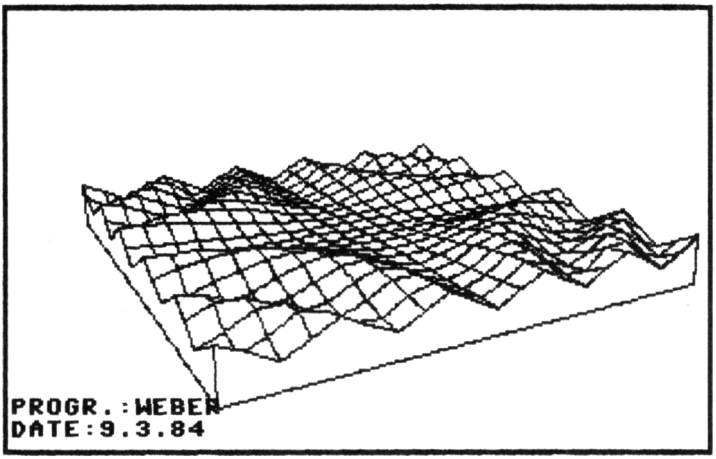
$$p = \sqrt{(x-r1)^2 + (y-r2)^2}$$

Eine Kreiswelle Könnten wir dann wie folgt definieren:

```
m = π/10
p = x^2 + y^2
z = 1-cos(m*5*p)
```

Durch die Verwendung von Hüllkurven, Kopplung rotationssymmetrischer Funktionen untereinander und mit anderen, Abhängigkeit der Position etc. kann man hier sehr interessante Effekte erzielen. Näheres in den Beispielprogrammen auf den nächsten Seiten.

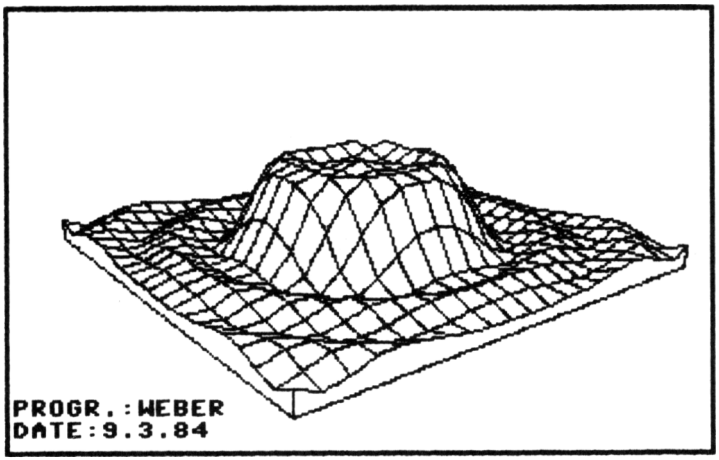
Beispiele zur Definition von 3D-Funktionen



Die verwendete Funktion lautet:

```
660 Z(1) = .5*(1-COS(5*M*X*Y/10)
```

Da die Variablen x und y von -10 bis +10 laufen ergeben sich als Extremwerte für x*y = +/- 100. Um eine halbwegs vernünftige Schrittweite im Argument des Kosinus zu erhalten, teilen wir diesen Wert durch 10, wodurch die Extremwerte zu +/- 10 werden. Dadurch können wir dann genau angeben, wieviele Perioden der Kosinus auf der Strecke (0/0)-(10/10) durchlaufen soll, nämlich genau 2.5. Deshalb verwenden wir 5*M, da M als $\pi/10$ definiert wurde. Die Werte des Argumentes laufen dann von 0 bis 5π, was exakt 2.5 Perioden des Kosinus entspricht.



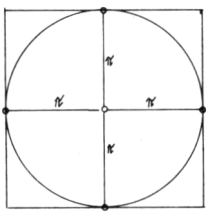
Verwendet wurde die Funktion:

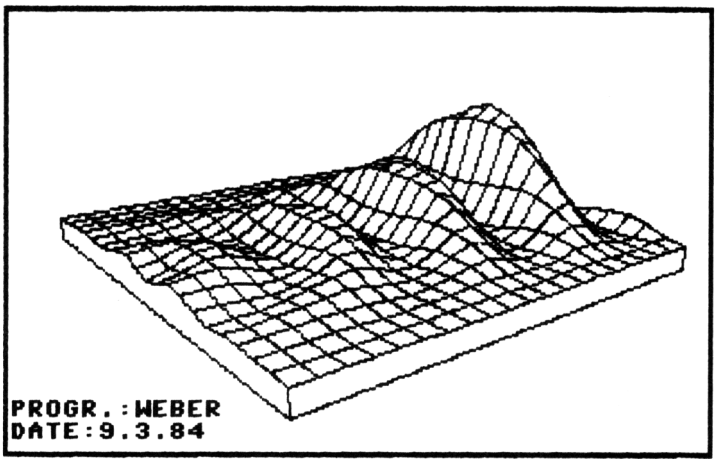
```
 650 P=SQR(X↑2+Y↑2)
 660
Z(1)=1.4+S*(COS(M*P)-COS(3*M*P)/3+COS(5*M*P)/5)
```

Durch Zeile 650 lassen wir die Funktion um (0/0) rotieren. Die Funktion in Zeile 660 ist die 3.Näherung einer Rechteckskurve durch Kosinusfunktionen (eine sog. Fouriertransformierte der Rechteckschwingung). Durch den Faktor M*P erhalten wir für die Koordinaten

$$\begin{bmatrix} 0 \\ 10 \end{bmatrix}, \begin{bmatrix} 0 \\ -10 \end{bmatrix}, \begin{bmatrix} 10 \\ 0 \end{bmatrix}, \begin{bmatrix} -10 \\ 0 \end{bmatrix}$$

den Berührungspunkten der Kreise um (0/0) mit dem Ausschnittsquadrat, jeweils den Wert π.





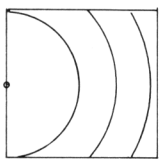
Als Funktion haben wir definiert:

```
650 P=SQR((X+10)↑2+Y↑2)
660 Z(1)=(1-COS(4*M*P)*EXP(X/8)*.5*(1+COS(M*Y))
```

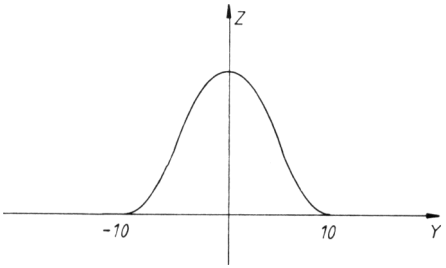
Dadurch, daß wir p als

$$p = \sqrt{(x+10)^2 + y^2}$$

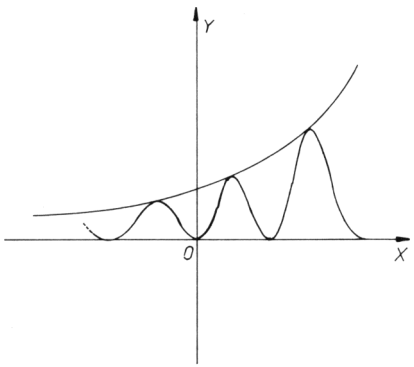
definieren, rotiert die Funktion nicht mehr um (0/0), sondern um (-10/0)

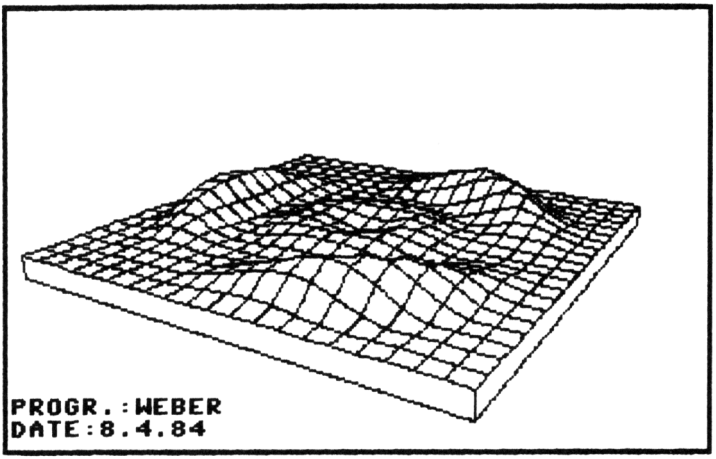


Die Kreiswelle (1-cos(4*m*p)) durchläuft im x-Bereich (-10;0) genau zwei Perioden, also auf (-10;10) 4 Perioden.
Durch Verwendung zweier Hüllkurven können wir die auf den Maxima erreichte Höhe, beziehungsweise den Verlauf der Funktion, ändern.
Durch die Multiplikation mit exp(x/8) erreichen wir ein exponentielles Ansteigen der Maxima mit exp(x/8) von links nach rechts.



Durch die HüllKurve (1-cos(m*y)) erreichen wir eine Einengung der Kurve in y-Richtung, da die HüllKurve, wie aus folgender SKizze ersichtlich, verläuft.





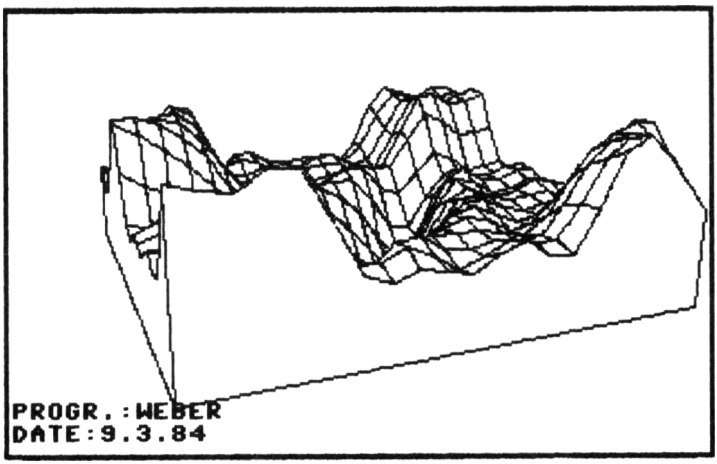
Die hierzu verwendete Funktion ist schon wesentlich komplizierter:

```
 650 P=SQR(X↑2+Y↑2)
 651 IFX=0THENW=0:GOTO659
 652 W=ATN(Y/X)- *(X<0)
 659 Z=-.5*(1-COS(Y-M*P))*(P<=10)
 660 Z(1)=-.8*(1-COS(3*W))*Z*(P>5)-.5*Z*(P<=5)
```

Unsere Grundfunktion ist eine Kreiswelle um (0/0) (Zeilen 650/659). Den zweiten Kreisring beeinflußen wir in Abhängigkeit von seiner Position im Koordinatensystem. Dazu bestimmen wir den Winkel w, den der Ortsvektor P mit der positiven x-Achse einschließt (Zeilen 651/652). Dies muß über den arctan(x) geschehen, da er als einziger in BASIC definiert ist. Da der arctan nur zwischen $-\pi/2$ und $\pi/2$ definiert ist, müßen wir, wenn x<0 ist π hinzu addieren. Dies geschieht durch $-\pi*(x<0)$.
Dabei mag ihnen der Ausdruck (x<0) als syntaktisch falsch erscheinen. Das ist aber nicht so. Der C-64 hat nämlich, was in keinem Bedienungshandbuch steht, die Boole'schen Wahrheitswerte definiert. Er setzt, falls der Ausdruck (x<0) zutrifft, an seiner Stelle den Wert -1, andernfalls den Wert 0. Durch $-\pi*(x<0)$ erhalten wir also

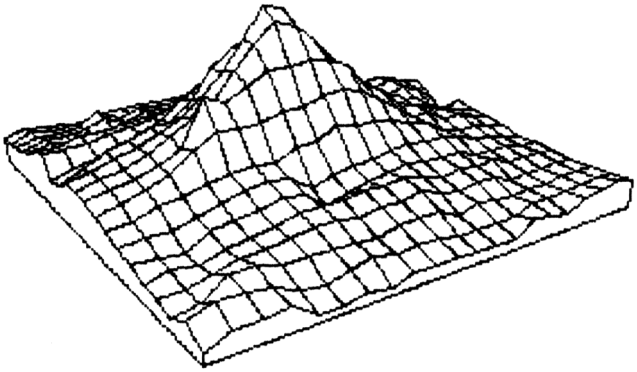
π ; x<0
0 ; x>=0

Auf dieselbe Art erreichen wir die Begrenzung der Beeinflussung der Kreiswelle auf den Bereich p>5 bzw. der Kreiswelle selbst auf den Bereich p<=10 (Zeilen659/660).
Durch (1-cos(3*w)) erzielen wir ein dreimaliges Maximun (bzw. Minimum) der Kreiswelle im Bereich p>5. Der Winkel zwischen den einzelnen Maxima beträgt dann genau 120°.



Diese Funktion ist ein Beispiel für die additive Überlagerung zweier Funktionen. Dabei wurde die Funktion aus Beispiel 2 um die Punkte (-19/-6) und (10/10) rotieren gelassen und die Einzelfunktionswerte addiert.

```
650 P1=SQR((X+10)↑2+(Y+6)↑2)
651 P2=SQR((X-10)↑2+(Y-10)↑2)
659 Z(1)=1.6+2*(COSM*P1)-COS(3*M*P1)/3
          +COS(5*M*P1)/5))
661 Z(1)=(Z(1)+2*(COS(M*P2)-COS(3*M*P2)/3
          +COS(5*M*P2)/5))
```



'Elevation modell programming' ist ein Ausdruck aus dem Amerikanischen und steht für die Möglichkeit mit Hilfe des Computers dreidimensionale Modelle von Landschaften zu entwerfen. Daß dies gar nicht so schwer ist haben wir ja im vorangegangenen Kapitel gesehen. Dieses Verfahren hat den Vorteil, daß viele Strukturen, die auf Karten mit Höhenlinien nur sehr schwer oder jedenfalls nur für das gübte Auge erkennbar sind, hier meistens sehr leicht ausgemacht werden können. Das primäre Problem der ganzen Angelegenheit ist gar nicht einmal das Programm selbst, ein wesentlich größeres Problem ist die Beschaffung von geeignetem Datenmaterial. Geeignet sind hierfür in erster Linie wiederum Karten mit Höhenlinien.

Als Beispielprogramm finden sie hier ein Programm, das sich auf Daten des U.S. Geological Survey über den Ausbruch des Mount St. Helen im Jahre 1980 stützt. Es ist hierfür lediglich notwendig ein paar geringfügige Modifikationen an unserem 3D-Programm vorzunehmen:

Am komfortabelsten ist es das erforderliche Datenmaterial am Ende des Programms in Form von Datazeilen abzuspeichern. Im Hauptprogramm tritt dann an Stelle der Berechnung der Funktionswerte die Zeile:

```
  READZ(1):Z(1)=Z(1)/7
```

Der Zusatz Z(1)=Z(1)/7 wird dadurch notwendig, da

es sich bei den Daten um Keine Originalhöhenangaben handelt, sondern um relative Werte. Durch den Zusatz werden die Daten in das richtige Verhältnis zur horizontalen Ausdehnung des Bildes gesetzt.

Die Reihenfolge der Daten ist so gewählt, daß sie von Norden her auf den Vulkan blicken. Wollen sie ihn von Süden sehen,so schreiben sie anstelle von:

```
FORY=-10TO10
FORX=-10TO10
```

die Zeilen

```
FORY=10TO-10STEP-1
FORX=10TO-10STEP-1
```

Die Daten werden dann so eingelesen, als wenn sie das Bild um 180 gedreht hätten.

Im folgenden finden sie das notwendige Datenmaterial für den AnblicK vor und nach dem Ausbruch in Form von Datazeilen.

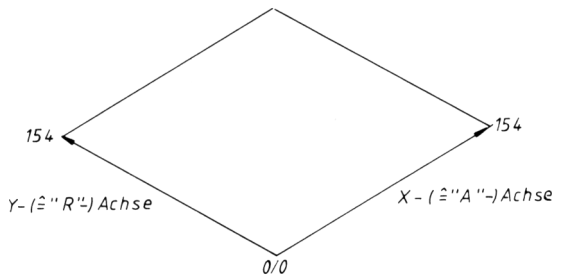
```
3000 DATA 8,10,9,8,5,6,7,3,3,3,4
          4,4,2,3,2,2,7,6,6,3
3001 DATA 8,8,8,7,7,7,5,5,5,5,6
          6,6,7,6,4,5,8,9,9,7
3002 DATA 8,8,8,7,7,7,7,8,6,9,9
          9,7,8,8,8,8,10,10,5,4
3003 DATA 8,9,8,8,8,10,10,9,10,12,11
          11,10,11,11,10,9,9,7,5,5
3004 DATA 9,10,10,10,12,11,12,11,12,14,13
          13,12,12,12,10,9,8,6,7,5
3005 DATA 10,10,11,11,11,13,13,13,14,13,12
          10,8,13,14,12,11,10,6,7,7
3006 DATA 10,11,12,13,15,16,16,17,16,15,14
          13,11,15,16,15,13,11,9,9,8
3007 DATA 12,13,14,14,15,19,20,22,23,23,24
          21,20,17,18,17,16,14,13,11,10
3008 DATA 13,14,16,16,19,19,23,27,30,32,31
          29,29,27,26,27,23,18,15,13,11
3009 DATA 14,15,16,18,20,22,27,30,37,39,40
          39,39,35,32,34,21,18,16,14,13
3010 DATA 15,14,16,17,21,22,26,31,37,44,49
          51,45,38,31,27,23,19,16,15,13
3011 DATA 8,13,17,17,18,21,26,34,41,43,53
          54,45,37,30,25,22,19,17,15,13
3012 DATA 14,17,19,18,19,23,27,34,39,42,46
          45,41,36,28,22,22,19,17,13,10
3013 DATA 18,19,20,19,20,23,27,30,33,36,37
          39,33,30,26,21,18,16,13,12,9
3014 DATA 16,15,20,19,21,23,24,26,28,27,29
          30,28,25,22,20,18,17,16,15,15
3015 DATA 16,17,15,16,20,19,21,22,21,21,22
          23,22,20,19,18,17,16,16,14,14
3016 DATA 11,12,10,13,19,16,17,17,17,17,17
          18,17,16,15,14,13,13,12,13,11
3017 DATA 5,6,10,13,18,16,16,16,17,16,16
          17,16,15,16,13,16,13,12,12,10
3018 DATA 11,15,19,20,22,19,15,16,15,15,16
          15,15,14,15,13,16,12,11,11,9
3019 DATA 12,13,15,13,11,15,13,13,13,13,13
          12,11,12,11,10,9,11,10,11,8
3020 DATA 10,14,18,14,13,13,14,14,13,13,13
          12,12,12,11,11,9,9,10,10,8
```

```
3000 DATA 8,10,9,8,5,6,7,3,3,3,4
          4,4,2,3,2,2,7,6,6,3
3001 DATA 8,8,8,7,7,7,5,5,5,5,6
          6,6,7,6,4,5,8,9,9,7
3002 DATA 8,8,8,7,7,7,7,8,6,9,9
          9,7,8,8,8,8,10,10,5,4
3003 DATA 8,9,8,8,8,10,10,9,10,12,11
          11,10,11,11,10,9,9,7,5,5
3004 DATA 9,10,10,10,12,11,12,11,12,14,13
          13,12,12,12,10,9,8,6,7,5
3005 DATA 10,10,11,11,11,13,13,13,14,13,12
          10,8,13,14,12,11,10,6,7,7
3006 DATA 10,11,12,13,15,16,16,17,16,15,14
          13,11,15,16,15,13,11,9,9,8
3007 DATA 12,13,14,14,15,19,20,22,23,23,24
          21,20,17,18,17,16,14,13,11,10
3008 DATA 13,14,16,16,19,19,23,27,30,32,31
          29,29,27,26,27,23,18,15,13,11
3009 DATA 14,15,16,18,20,22,27,30,37,39,41
          41,39,35,32,34,21,18,16,14,13
3010 DATA 15,14,16,17,21,22,26,31,37,44,40
          45,44,38,31,27,23,19,16,15,13
3011 DATA 8,13,17,17,18,21,26,34,41,37,30
          30,40,38,30,25,22,19,17,15,13
3012 DATA 14,17,19,18,19,23,27,34,40,32,28
          29,40,36,28,22,22,19,17,13,10
3013 DATA 18,19,20,19,20,23,27,30,33,31,26
          28,34,31,26,21,18,16,13,12,9
3014 DATA 16,15,20,19,21,23,24,26,28,27,24
          24,25,25,22,20,18,17,16,15,15
3015 DATA 16,17,15,16,20,19,21,22,21,21,22
          23,22,20,19,18,17,16,16,14,14
3016 DATA 11,12,10,13,19,16,17,17,17,17,17
          18,17,16,15,14,13,13,12,13,11
3017 DATA 5,6,10,13,18,16,16,16,17,16,16
          17,16,15,16,13,16,13,12,12,10
3018 DATA 11,15,19,20,22,19,15,16,15,15,16
          15,15,14,15,13,16,12,11,11,9
3019 DATA 12,13,15,13,11,15,13,13,13,13,13
          12,11,12,11,10,9,11,10,11,8
3020 DATA 10,14,18,14,13,13,14,14,13,13,13
          12,12,12,11,11,9,9,10,10,8
```

3D - Die zweite Version

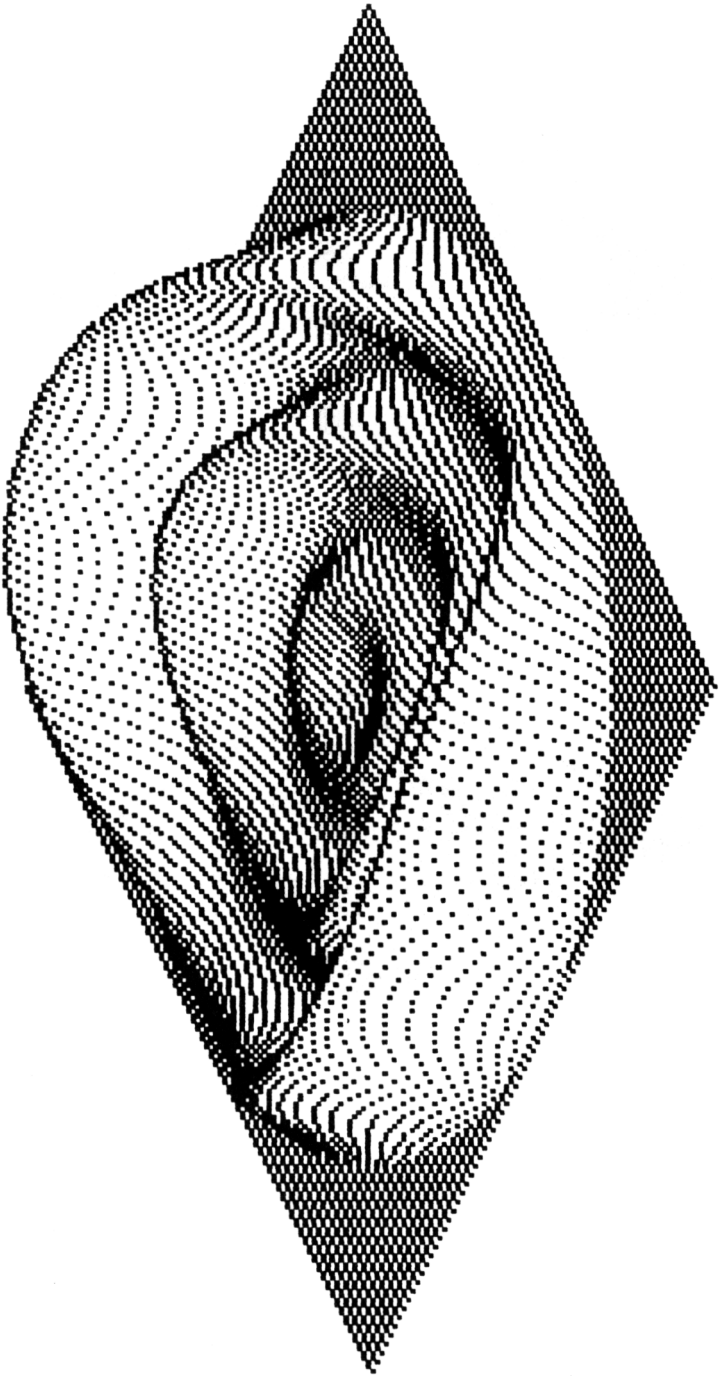
Das folgende Programm liefert sehr hochaufgelöste dreidimensionale Funktionsplots, deren Qualität vor allem durch die ausgeprägte Licht-Schattenwirkung, die dabei entsteht, überzeugt. Es ist außerdem wesentlich einfacher als das vorher bereits beschriebene Verfahren, aber, und das ist das größte Handicap dieses Programms, es braucht unvergleichlich länger um einen identischen Funktionsplot zu erstellen. Gehen wir davon aus, daß es sich nicht um eine ganz einfache Funktion handelt, so benötigt das Programm zur Berechnung im Schnitt zwischen 2 und 6 Stunden (!). Es berechnet in diesem Zeitraum 11.858 Einzelpunkte (!). Wir gehen deshalb nicht besonders intensiv auf dieses Programm ein. Am besten, sie spielen sich selbst etwas mit der Definition eigener Funktionen. Sie werden dann bald bemerken wie sich z.B. eine zusätzliche, ins Progarmm aufgenommene SQR-Routine auswirkt. Sie erzielen damit nämlich eine Verlängerung der Laufzeit um gut eine halbe Stunde. Dies liegt in erster Linie daran, daß die SQR-Routine des Commodore BASIC extrem langsam konvergiert, und, das nebenbei gesagt, noch nicht einmal besonders gut.
Deshalb an dieser Stelle nur ein paar Worte zum Aufbau des Grundquadrats:

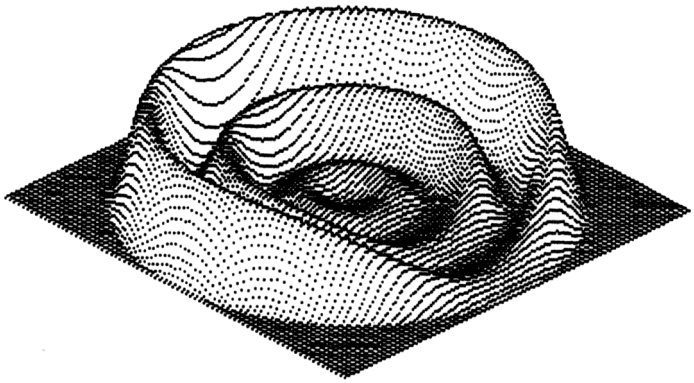
Es handelt sich um eine Parallelprojektion, wobei wir unter 30 von oben auf die Grundebene blicken. Das Programm rechnet nach folgendem Schema:

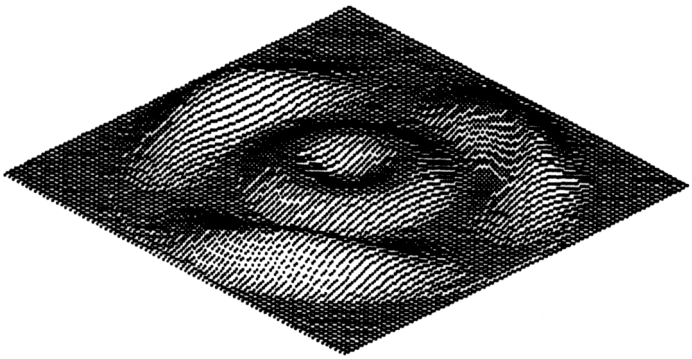


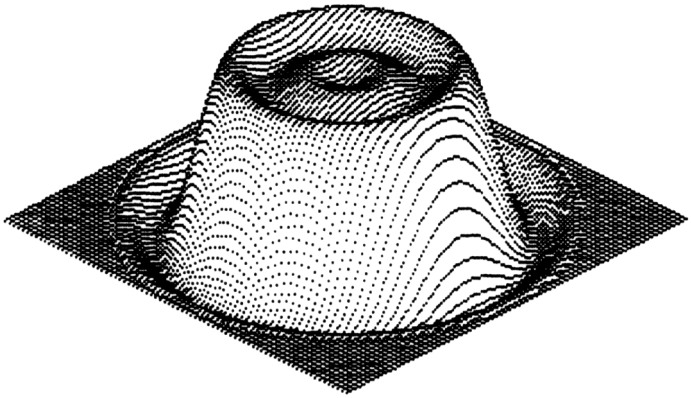
Der Mittelpunkt liegt bei A=77,R=77 (vgl. Zeile 160-180). Wichtig ist der Faktor 10 in Zeile 170 zur Amplitudenvergrößerung, da sie sonst kaum etwas sehen dürften. Diese Amplitudenvergrößerung

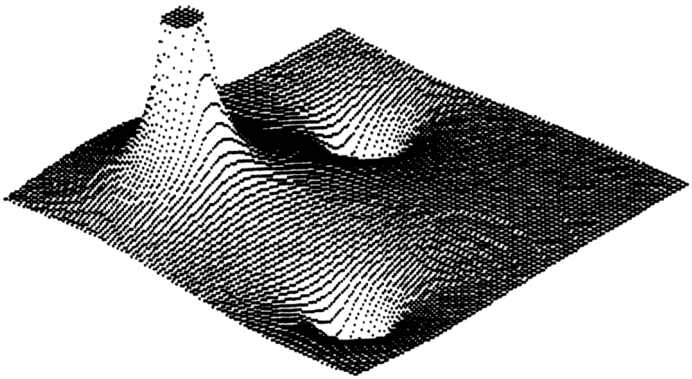
ist dadurch notwendig geworden, daß das Programm ohne Projektionsformel rechnet.Wir haben in Teil 1 diese Tatsache schon einmal kurz angeschnitten. Die Parallelprojektion ist eine lineare Abbildung, was bei der Zentralprojektion ja nicht der Fall ist. Es genügt uns daher vollständig die Bilder der Basisvektoren $\overline{x}'$,$\overline{y}'$,$\overline{z}'$ unserer Originalebene zu bestimmen, da sie eine Basis unseres Bildraumes, sprich des Bildschirmkoordinatensystems darstellen.
Wir haben die Funktionen in Abhängigkeit von den speziellen Bildschirmvariablen A und X definiert und müßen deshalb eine bestimmte Vergrößerung herstellen, z.B. durch den Faktor 10. Der korrekte Wert wäre $10*\sqrt{2}*\cos 30^\circ = 12.2$.
Auf den folgenden Seiten sehen sie ein paar Beispiele für mit diesem Verfahren erzeugte Funktionen.



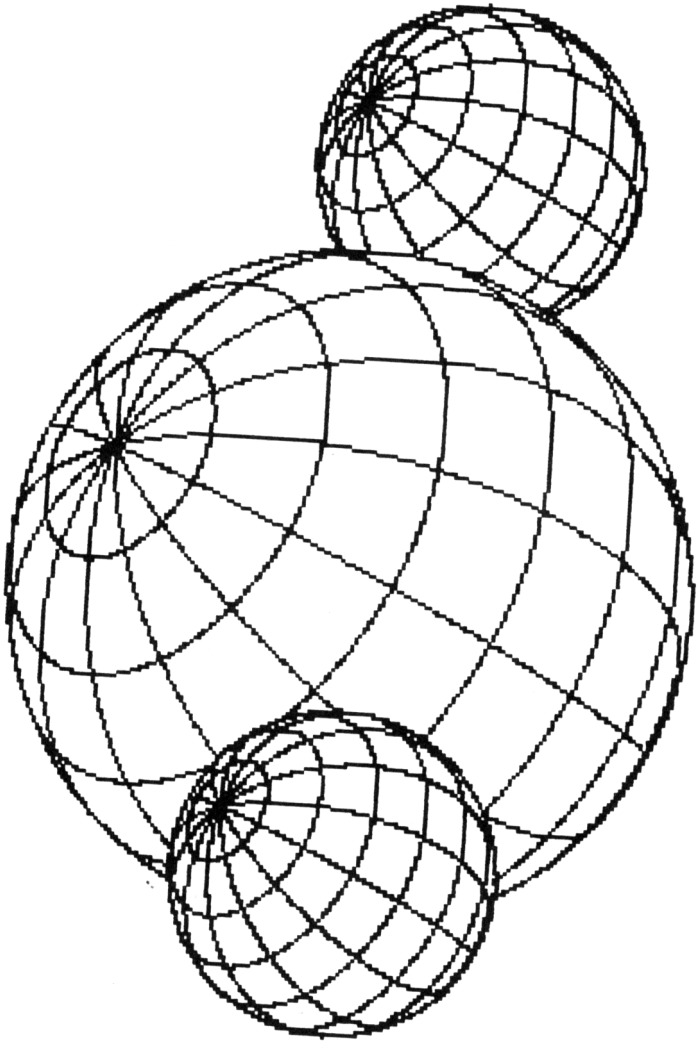








Dreidimensionale Kugeln



Um Kugeln dreidimensional darstellen zu können, überlegen wir uns zunächst, wie man eine Kugel im R^3 beschreibt.
Eine Kugel sind diejenigen Orte, die einen bestimmten Abstand r von einem vorgegebenen Punkt haben, z.B. dem Ursprung O. Das heißt: Lassen wir den vorgegebenen Radiusvektor $\vec{r}$ beliebig um O

rotieren, so liegt seine Spitze immer auf einer Kugel. Veranschaulichen wir uns das zunächst an der Skizze auf der gegenüberliegenden Seite.

Wenn wir eine Breitenkreis beschreiben wollen, dann lassen wir die z-Koordinate konstant und den Restvektor um die z-Achse rotieren: Nach (1) gilt:

zr = rsin(i)

Ein Kugelpunkt P läßt sich vektoriell schreiben als $\overline{r}=\overline{xr}+\overline{yr}+\overline{zr}$. Der Vektor $\overline{xr}+\overline{yr}$ liegt in der x,y-Ebene. Für ihn gilt: $|\overline{xr}+\overline{yr}|=|rcos(i)|$.
Nach (2) gilt nun für xr und yr

xr = (rcos(i))cos(g)
yr = (rcos(i))sin(g)

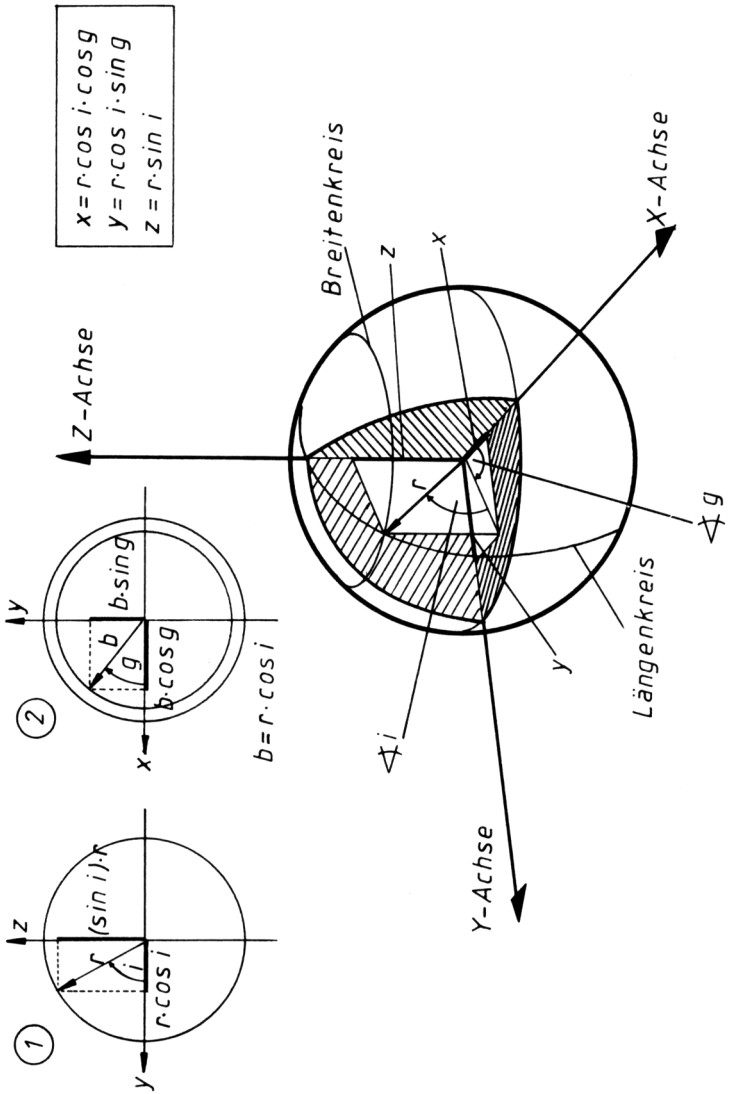
Wir haben damit die sog. Polarkoordinatendarstellung eines Kugelpunktes $\overline{P}=(r,i,g)$ gefunden. Für ihn gelten die Gleichungen:

$$\overline{P} = (r,i,g) = \begin{bmatrix} xr \\ yr \\ zr \end{bmatrix} = \begin{bmatrix} rcos(i)cos(g) \\ rcos(i)sin(g) \\ rcin(i) \end{bmatrix}$$

Zurück zu unserem Breitenkreis: Wir erhalten den Breitenkreis dadurch, daß wird den Wert i konstant lassen, den Wert g aber von 0 bis $2*\pi$ laufen lassen. Wir erhalten damit den Breitenkreis von i.

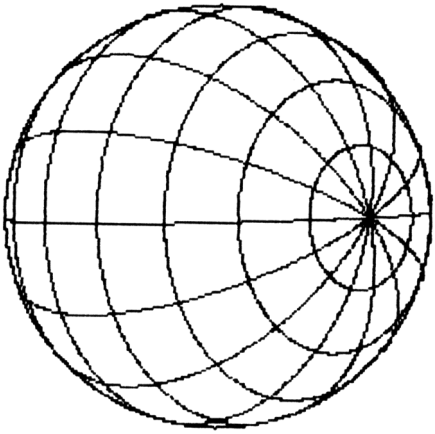
Die Darstellung eines Längenkreises läuft völlig analog. Hier lassen wir diesmal g konstant und i von 0 bis $2*\pi$ laufen.
Damit haben wir das schwierigste schon erledigt.Wenn wir die Koordinatendarstellung eines Punktes $\overline{P}=(xr/yr/zr)$ ermittelt haben,können wir diesen Vektor mit Hilfe der Rotationsmatrix beliebig drehen. Wenn wir das mit allen Punkten der Kugel machen, drehen wir also die Kugel.
Zur Darstellung auf dem Bildschirm verwenden wir sinnvollerweise die Parallelprojektion parallel zur y-Achse in die x,z-Ebene. Dann ist ein Punkt der Oberfläche offensichtlich genau dann unsichtbar, wenn seine yr-Koordinate negativ ist. Damit haben wir eine einfache Methode zur Unterdrückung verdeckter Linien gefunden. Auf die Projektionsformel können wir hier verzichten, da durch die Projektionsrichtung parallel zur y-Achse die y-Koordnate 0, xr und zr aber erhalten bleiben. Wir verwenden dann einfach xr

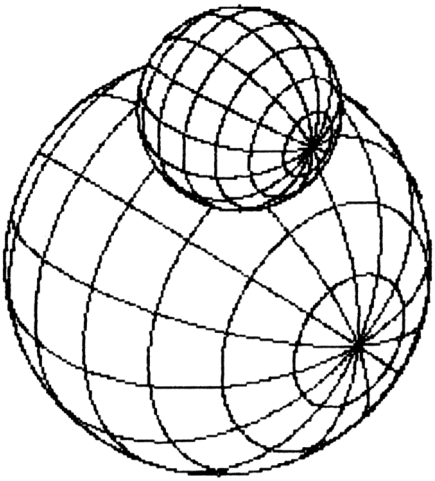
und zr als zweidimensionale Bildschirmkoordinaten. Das folgende Programm ist nach diesen Überlegungen aufgebaut.
Sie können dieses Verfahren nun vielfältig nutzen. Um nur ein paar Beispiele zu nennen: Bei der Eingabe entsprechend vieler Koordinaten können sie sich einen Globus aus jeder beliebigen Position betrachten. Sie können damit Modelle einfacher Moleküle, wie etwa das Wassermolekül H_2O oder das Ammoniakmolekül NH_3 herstellen.

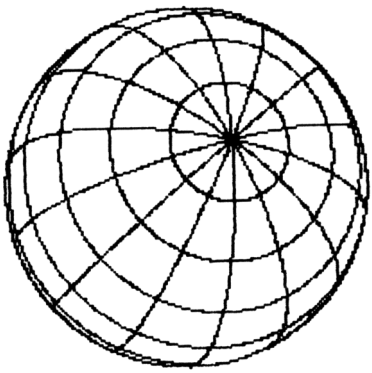


```
100 ****************************
110 * DREIDIMENSIONALE KUGELN *
120 ****************************
130 :
140 INPUT"␣W1,W2,W3";W1,W2,W3
150 £0:£1:£L14,6
160 W1=W1*π/180:W2=W2*π/180:W3=W3*π/180
170 :
180 ************************************
190 * BERECHNEN DER ROTATIONSMATRIX *
200 ************************************
210 :
220 GOSUB560:£G160,100,95
230 :
240 ******************
250 * LAENGENKREISE *
260 ******************
270 :
280 FORG=0TO2πSTEPπ/7:FORI=0TO2*πSTEP.01
290 Z=SIN(I):X=COS(I)*COS(G):Y=COS(I)*SIN(G)
300 XA=M(1,1)*X+M(2,1)*Y+M(3,1)*Z
310 YA=M(1,2)*X+M(2,2)*Y+M(3,2)*Z
320 ZA=M(1,3)*X+M(2,3)*Y+M(3,3)*Z
330 IFYA<0THEN360
340 XB=160+95*XA:YB=100+95*ZA
350 £QXB,YB
360 NEXT:NEXT
370 :
```

```
380 ******************
390 * BREITENKREISE *
400 ******************
410 :
420 FORI=(-π/2+π/9)TO(π/2-π/9)STEPπ/9:FORG=0TO2*πSTEP.01
430 Z=SIN(I):X=COS(I)*COS(G):Y=COS(I)*SIN(G)
440 XA=M(1,1)*X+M(2,1)*Y+M(3,1)*Z
450 YA=M(1,2)*X+M(2,2)*Y+M(3,2)*Z
460 ZA=M(1,3)*X+M(2,3)*Y+M(3,3)*Z
470 IFYA<0THEN360
480 XB=160+95*XA:YB=100+95*ZA
490 £QXB,YB
500 NEXT:NEXT:END
510 :
520 ********************
530 * ROTATIONSMATRIX *
540 ********************
550 :
560 M(1,1)=COS(W2)*COS(W3)
570 M(2,1)=-COS(W2)*SIN(W3)
580 M(3,1)=SIN(W2)
590 M(1,2)=COS(W1)*SIN(W3)+SIN(W1)*SIN(W2)*COS(W3)
600 M(2,2)=COS(W1)*COS(W3)-SIN(W1)*SIN(W2)*SIN(W3)
610 M(3,2)=-SIN(W1)*COS(W2)
620 M(1,3)=SIN(W1)*SIN(W3)-COS(W1)*SIN(W2)*COS(W3)
630 M(2,3)=SIN(W1)*COS(W3)+COS(W1)*SIN(W2)*SIN(W3)
640 M(3,3)=COS(W1)*COS(W2)
650 RETURN

READY.
```

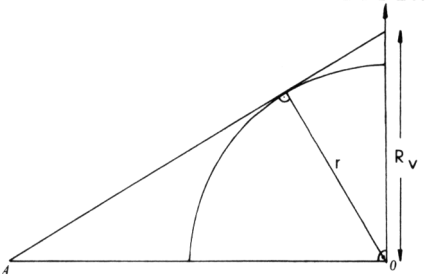






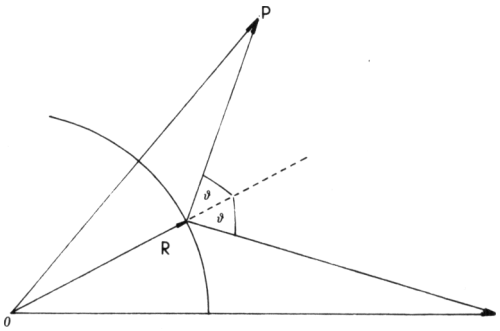
Kugelspiegelungen

Eine schöne Anwendung der Vektorrechnung erhalten wir, wenn wir die Spiegelbilder auf einer Seifenblase oder Metallkugel berechnen wollen. Wir gehen hierbei davon aus, daß das Auge seinen Standpunkt unveränderlich bei (0/10/0) beibehält und auf eine Kugel mit Radius 1 und Mittelpunkt im Ursprung blickt. Das Bild, das sich dem Betrachter hier bietet, erhalten wir wieder durch die bereits bekannte Form der Zentralprojektion, indem wir den Sehstrahl mit xz-Ebene schneiden. Wie aus untenstehender Abbildung ersichtlich hat die Kugel dabei einen virtuellen Radius von $1/\sqrt{1-1/|A|}$.



Wollen wir nun den Punkt der Kugel ermitteln, auf dem das Spiegelbild eines vorgegebenen Punktes P für den Betrachter sichtbar wird, so gehen wir wie folgt vor:
Der Punkt liegt bestimmt auf dem Schnittkreis der Kugel mit der durch die Vektoren $\overline{P}$ und $\overline{A}$ aufgespannten Ebene. Weiter wissen wir, daß sein Ortsvektor $\overline{R}$ zwischen $\overline{P}$ und $\overline{A}$ liegen muß. Ausgehend vom Spiegelgesetz für ebene Spiegel, bei dem ja bekanntlich gilt, Einfallswinkel ist gleich Ausfallswinkel, kommen wir auf folgende Beziehung, die von $\overline{R}$ erfüllt werden muß:

$$\frac{\overline{R} \circ \overline{RA}}{|\overline{R}|\,|\overline{RA}|} = \frac{\overline{RP} \circ \overline{R}}{|\overline{R}|\,|\overline{RP}|}$$



Das folgende Programm bestimmt den Punkt $\overline{R}$, der diese Bedingung erfüllt iterativ. Als erste Näherung wird hierfür die Winkelhalbierende von $\overline{A}$ und $\overline{P}$ verwendet. Das Verfahren ist nach spätestens 3 Iterationen auf weniger als 0.1° genau.

Für die Iteration selbst legen wir in die $\overline{A}\overline{P}$-Ebene ein kartesisches Koordinatensystem mit dem Vektor $\overline{A}$ als x-Achse. Die y-Achse ermitteln wir durch zweimaliges Kreuzprodukt in der Form $(\overline{A}x\overline{P})x\overline{A}$. Innerhalb dieser Ebene ist die Iteration wesentlich einfacher, da wir, wenn wir den Winkel w zwischen $\overline{A}$ und $\overline{P}$ kennen, den winkelhalbierenden Radiusvektor durch

$$\overline{R} = \begin{bmatrix} r\ \cos(w/2) \\ r\ \sin(w/2) \end{bmatrix}$$

erhalten.
Wir bestimmen nach der oben bereits angesprochenen Formel die Cosini der Winkel w1 zwischen $\overline{RP}$ und $\overline{R}$ und w2 zwischen $\overline{R}$ und $\overline{RA}$. Im Falle w1<w2 wird der Winkel w/2 um den Wert w/4 reduziert, im anderen Fall um den Wert w/4 erhöht. Beim nächsten Durchgang dann um w/8, w/16 usw.. Die Iteration unterschreitet dann spätestens beim 4 Iterationsschritt die Genauigkeitsschwelle von 0.1.

Da wir natürlich Punkte, die hinter der Kugel liegen, nicht sehen können, bestimmen wir einfach im Falle p2<0 als erstes den Schnittpunkt der Verbindungsgeraden $\overline{PA}$ mit der xz-Ebene. Ist sein Abstand von 0 kleiner als der virtuelle Radius RV, so können wir den Punkt gleich vergessen.

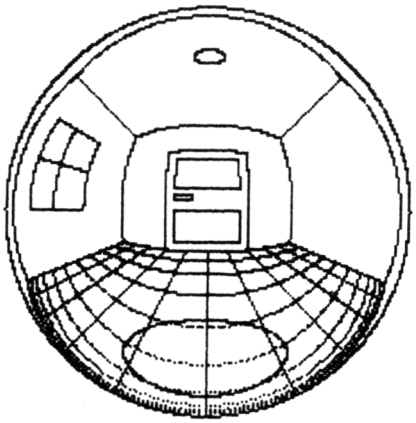
Als Beipiel sehen sie eine MetallKugel,die sich freischwebend in der Mitte eines Raumes befindet. Mit auf dem Bild sind die Lampe an der Decke sowie der Schatten, den die Kugel infolge der Bestrahlung wirft.

```
100 **********************
110 * KUGELSPIEGELUNGEN *
120 **********************
130 :
140 £0:£1:£L11,7:MX=200:MY=100:MO=30
150 A1=0:A2=10:A3=0
160 GOSUB950
170 :
180 ************
190 * UMKREIS *
200 ************
210 :
220 A1=0:A2=10:A3=0
230 BA=SQR(A1↑2+A2↑2+A3↑3)
240 X1=A1/BA:X2=A2/BA:X3=A3/BA
250 RV=1/SQR(1-1/BA↑2)
260 :
270 £GMX,MY,MO*RV+1
280 GOTO860
290 :
300 ******************************
310 * KARTHESISCHE KOORDINATEN *
320 ******************************
330 :
340 B1=A3*(A3*P1-A1*P3)-A2*(A1*P2-A2*P1)
350 B2=A1*(A1*P2-A2*P1)-A3*(A2*P3-A3*P2)
360 B3=A2*(A2*P3-A3*P2)-A1*(A3*P1-A1*P3)
370 BB=SQR(B1↑2+B2↑2+B3↑2)
380 :
```

```
390 ****************************************
400 * TRANSFORMATION IN EBENENKOORDINATEN *
410 ****************************************
420 :
430 PX=P1*X1+P2*X2+P3*X3
440 PY=P1*Y1+P2*Y2+P3*Y3
450 AX=BA:AY=0
460 :
470 ********************
480 * WINKEL W (<P;A) *
490 ********************
500 :
510 BP=SQR(P1↑2+P2↑2+P3↑2)
520 QQ=(PX*AX+PY*AY)/(BP*BA)
530 W=-ATN(QQ/SQR(1-QQ↑2))+π/2:A=W/2:G=A
540 :
550 ************************
560 * COS(W1) UND COS(W2) *
570 ************************
580 :
590 RX=COS(A):RY=SIN(A)
600 T1=SQR((PX-RX)↑2+(PY-RY)↑2)
610 T2=SQR((AX-RX)↑2+(AY-RY)↑2)
620 W1=(PX-RX)*RX+(PY-RY)*RY
630 W2=(AX-RX)*RX+(AY-RY)*RY
640 IF(W1-W2)<.01THEN720
650 IFW1>W2THENA=A+(W1-W2)/2:GOTO590
660 IFW2>W1THENA=A-(W2-W1)/2:GOTO590
670 :
```

```
680 ***********
690 * AUSGABE *
700 ***********
710 :
720 G1=COS(A)*X1+SIN(A)*Y1
730 G2=COS(A)*X2+SIN(A)*Y2
740 G3=COS(A)*X3+SIN(A)*Y3
750 IFG2-A2=0THENT=0:GOTO770
760 T=G2/(G2-A2)
770 GX=G1-T*(G1-A1)
780 GY=G3-T*(G3-A3)
790 £QMX+MO*GX,MY-MO*GY
800 RETURN
810 :
820 **********************
830 * FIGURENDEFINITION *
840 **********************
850 :
860 P3=2:FORP1=-1TO5STEP.1:P2=-1:GOSUB310:F=GX:R=GY:P2=5:GOSUB310
870 REM £EMX+MO*F,MY-30*R,MX+MO*GX,MY-MO*GY:NEXT
880 NEXT
890 P3=2:FORP2=-1TO5STEP.1:P1=5:GOSUB310:F=GX:R=GY:P1=-1:GOSUB310
900 REM  £EMX+MO*F,MY-30*R,MX+MO*GX,MY-MO*GY:NEXT
910 NEXT
920 REMP3=5:P2=5:FORP1=5TO1STEP-.1:GOSUB21:NEXT
930 REM P1=1:P3=5:FORP2=5TO-1STEP-.1:GOSUB21:NEXT
940 END
950 P3=-1:FORX=-3.2TO1.8STEP.5
960 P2=5:P1=X:GOSUB1040:X1=MX+30*PX:Y1=MY-30*PY
970 P2=-5:P1=X:GOSUB1040:X2=MX+30*PX:Y2=MY-30*PY
```

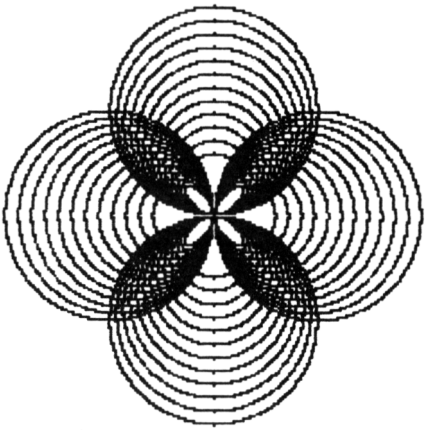
```
980 £EX1,Y1,X2,Y2
990 NEXT
1000 FORY=-5TO5STEP1
1010 P1=1.8:P2=Y:GOSUB1040:X1=MX+30*PX:Y1=MY-30*PY
1020 P1=-3.2:P2=Y:GOSUB1040:X2=MX+30*PX:Y2=MY-30*PY
1030 £EX1,Y1,X2,Y2:NEXT:RETURN
1040 IFP2-A2=0THENT=0:GOTO1060
1050 T=P2/(P2-A2)
1060 PX=P1-T*(P1-A1)
1070 PY=P3-T*(P3-A3)
1080 RETURN

READY.
```

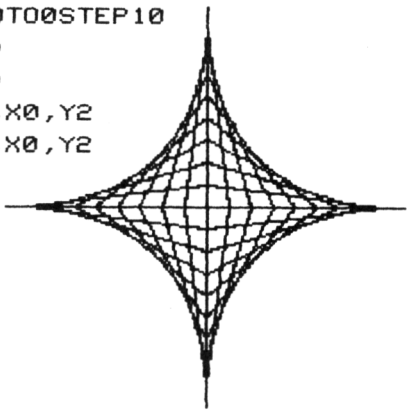
Anhang A

Sie finden auf den folgenden Seiten ein paar Beispiele für einfache Grafikprogramme, die aber eine nicht unbeträchtliche Wirkung erzielen.

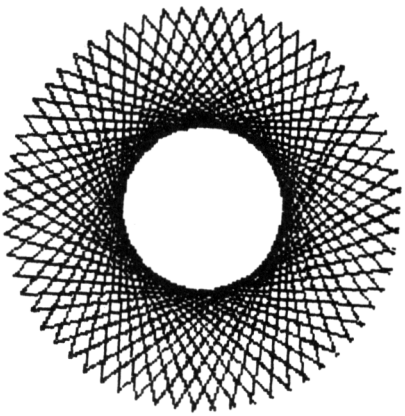
```
100 £0:£1:£L14,6
110 FORR=14TO47STEP3
120 £G160+R,100,R
130 £G160-R,100,R
140 £G160,100+R,R
150 £G160,100-R,R
160 NEXT
170 GETA$:IFA$=""THEN170
180 £2

READY.
```



```
100 £0:£1:£L14,11
110 X0=160:Y0=100:FORA=-90TO0STEP10
120 X1=160+A:Y1=100+(90+A)
130 X2=160-A:Y2=100-(90+A)
140 £EX1,Y0,X0,Y1:£EX1,Y0,X0,Y2
150 £EX2,Y0,X0,Y1:£EX2,Y0,X0,Y2
160 NEXT

READY.
```

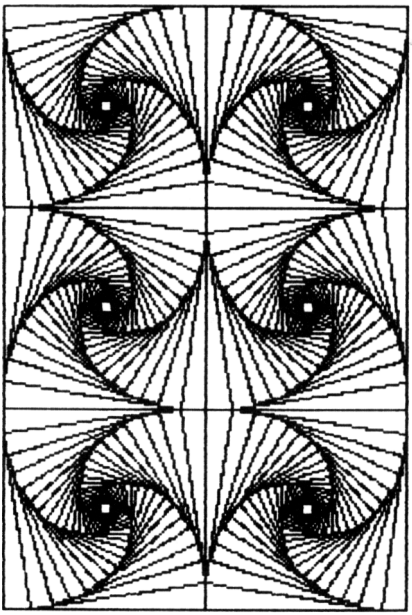


```
100 £0:£1:£L14,6
110 FORI=0TO2*πSTEP.1
120 X1=160+90*SIN(I):Y1=100+90*COS(I)
130 X2=160+90*SIN(I+2.3):Y2=100+90*COS(I+2.3)
140 £EX1,Y1,X2,Y2
150 NEXT

READY.
```



```
100 ************
110 * QUADRATE *
120 ************
130 :
140 £0:£1:£L14,6
150 R=0:GOTO390
160 :
170 ************************************
180 * BERECHNUNG DER INNEREN QUADRATE *
190 ************************************
200 :
210 READR:FORA=1TO23
220 FORI=1TO4
230 X(5)=X(1):Y(5)=Y(1)
240 XN(I)=(X(I+1)-X(I))/8+X(I)
250 YN(I)=(Y(I+1)-Y(I))/8+Y(I)
260 NEXT
270 FORI=1TO4:X(I)=XN(I):Y(I)=YN(I):NEXT
280 £EX(2),Y(2),X(1),Y(1)
290 £EX(3),Y(3),X(2),Y(2)
300 £EX(4),Y(4),X(3),Y(3)
310 £EX(1),Y(1),X(4),Y(4)
320 NEXT
330 IFR=2THEN500
340 :
350 ********************
360 * AUSGANGSQUADRATE *
370 ********************
380 :
390 FORI=1TO4:READX(I):READY(I):NEXT
```

```
400 £EX(2),Y(2),X(1),Y(1)
410 £EX(3),Y(3),X(2),Y(2)
420 £EX(4),Y(4),X(3),Y(3)
430 £EX(1),Y(1),X(4),Y(4)
440 GOTO210
450 :
460 ************************************
470 * KOORDINATEN DER AUSGANGSQUADRATE *
480 ************************************
490 :
500 DATA25,10,115,10,115,100,25,100,0
510 DATA115,100,205,100,205,10,115,10,0
520 DATA205,10,295,10,295,100,205,100,0
530 DATA25,100,25,190,115,190,115,100,0
540 DATA115,100,205,100,205,190,115,190,0
550 DATA205,190,295,190,295,100,205,100,2
560 GETA$:IFA$=""THEN560
570 £2

READY.
```



Anhang B

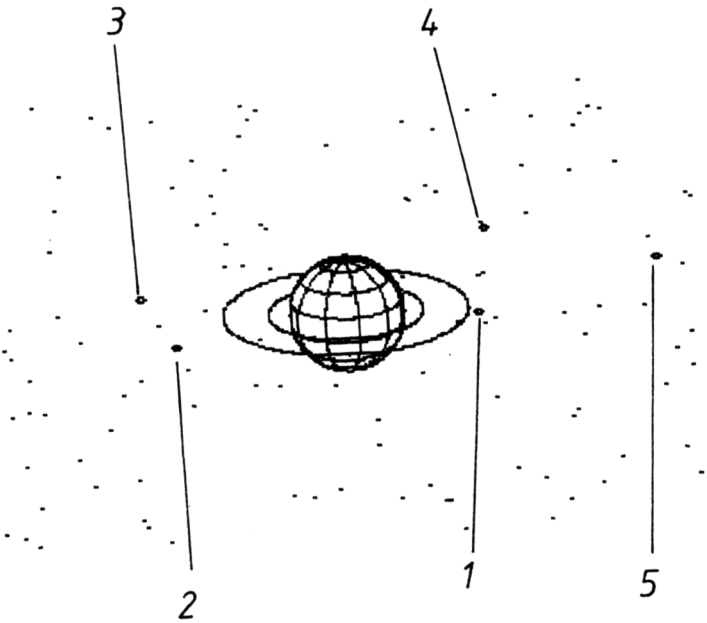
Das folgende Programm simuliert die Bewegung der Begleitmonde eines Planeten. Das Bild des Zentralplaneten wurde dabei mit Hilfe des Kugelprogramms erzeugt. Die Neigung der BlickRichtung gegenüber der Bahnebene beträgt 20.. Bei Vorgabe der Rotationsdauer des innersten Mondes kann die größere Periode der äußeren Monde durch die Division durch einen bestimmten Wert im Argument der Sinus- bzw. Kosinusfunktion erreicht werden. Bei frei gewähltem Bahnradius r ist die große Halbachse a der Ellipsenbahn auf dem Bildschirm=r, die kleine Halbachse b=r*sin(20). Die Verschiebung des Spritepuffers von 832 nach 17408 ist deshalb notwendig geworden, da der GrafikBildschirm des IWT Sprite Komfort Kit bei 8€000, und nicht wie ursprünglich bei 8€2000 angesiedelt ist. Das Sprite selbst ist in der linken oberen Ecke der Spritematrix definiert. Die Bildschirmmitte eine Sprites im Hiresbildschirm ist (182/152). Durch die Zeile

```
IF D1 < 25 AND Y < 152 ...
```

ermitteln wir wann der innerste Mond hinter dem Planeten unsichtbar wird. Der Radius des Zentralplaneten beträgt dabei 25 Bildschirmeinheiten.

```
100 £0:£1:£4"PLANET",8
110 :
120 * SPRITE DEFINIEREN *
130 :
140 FORI=832TO832+63:POKEI,0:NEXT:POKE832,96:POKE835,240:POKE838,240:POKE841,96
150 M1=182:M2=148
160 :
170 * VERSCHIEBEN *
180 :
190 FORI=0TO63:POKE17408+I,PEEK(832+I):NEXT
200 FORN=1TO5:£PN,13
210 £IN:£CN,7:NEXT
220 :
230 * STERNE *
240 :
250 FORI=1TO120:X=318*RND(1):Y=199*RND(1)
260 IFSQR((X-160)↑2+(Y-100)↑2)<25 THEN280
270 £QX,Y:£QX+1,Y
280 NEXT
290 :
300 * SYSTEM *
310 :
320 FORI=0TO40*πSTEP.5
330 * MOND 1 *
340 X1=M1+60*SIN(I/2)
350 Y1=M2+20*COS(I/2)
360 D1=SQR((X1-M1)↑2+(Y1-M2)↑2)
370 IFD1<25ANDY1<M2THEN:£01:£K1,X1,Y1:GOTO390
380 £I1:£K1,X1,Y1
390 * MOND 2 *
```

```
400 X2=M1+85*SIN( I/3)
410 Y2=M2+29*COS( I/3)
420 £K2,X2,Y2
430 * MOND 3 *
440 X3=M1+120*SIN( I/6)
450 Y3=M2+41*COS( I/6)
460 £K3,X3,Y3
470 * MOND 4 *
480 X4=M1+95*SIN( I/3.5)
490 Y4=M2+33*COS( I/3.5)
500 £K4,X4,Y4
510 * MOND 5 *
520 X5=M1+150*SIN( I/8)
530 Y5=M2+51*COS( I/8)
540 £K5,X5,Y5
550 NEXT

READY.
```


3
4
2
1
5

Anhang C

Auszug aus dem Befehlslisting des IWT Sprite Komfort Kit

↑ 0	GrafiK Clear
↑ 1	GrafiK ein
↑ 2	GrafiK aus
↑ La,b	Hintergrundfarbe a,Linienfarbe b
↑ Qx,y	PunKt an (x/y)
↑ Ex1,y1,x2,y2	Linie von (x1/y1)nach (x2/y2)
↑ Tx1,y1,x2,y2	Löscht Linie
↑ Bx1.y1,x2,y2	Rahmen mit den EcKpunKten (x1/y1)und (x2/y2)

Anhang D

Sie finden hier ein Maschinenprogramm, das es Ihnen erheblich erleichtert die Grafik anzusteuern. Das Programm wird dabei durch ein BASIC-Programm in den für Maschinensprache reservierten Speicherbereich ab $C000 (=49152) geschrieben. Dies ermöglicht es Ihnen das Programm auf jeden Fall zu verwenden, auch wenn sie nicht auf einen Assembler oder Monitor zurückgreifen können. Sollten sie jedoch im Besitz eines solchen sein, so finden sie im Anschluss ein Monitorlisting des gleichen Programms, das eine Programmierung natürlich wesentlich vereinfacht, da hierbei die Wahrscheinlichkeit einen Tippfehler zu begehen, wesentlich geringer ist.
Sie können diese Maschinenroutine nun wie folgt aufrufen:

SYS49152

PF = Punktfarbe
HF = Hintergrundfarbe

ŁO0,PF*16+HF	Grafik einschalten mit HIRES-Bildschirm löschen
ŁO1,PF*16+HF	Grafik einschalten ohne HIRES-Bildschirm löschen
ŁQX,Y	Setzt Punkt an Position (X/Y) im HIRES-Bildschirm
ŁCX,Y	Löscht Punkt an Position (X/Y)

Sämtliche Zusatzbefehle sind sowohl im Direkt- als auch im Programm-Modus lauffähig.

Im Anschluß daran finden sie ein Grafikunterprogramm ab BASIC-Adresse 10000, das die Funktionen Circle, Line und Box enthält. Der Aufruf durch das Basicprogramm erfolgt folgendermaßen:

```
AA=1:X1=..:Y1=..:X2=..:Y2=..:GOSUB10000
     LINIE (X1/Y1) - (X2/Y2)
AA=2:MX=..:MY=..:R=..:GOSUB10000
     KREIS UM (MX/MY) MIT RADIUS R
AA=3:X1=..:Y1=..:X2=..:Y2=..:GOSUB10000
     BOX (X1/Y1) - (X2/Y2)
```

Benötigte Speicherplätze und Subroutine-Erklärung:

$033C	Y -Koordinate
$14	XL-Koordinate
$15	XH-Koordinate
$0341	Y AND 7
$0342	INT(X/8)
$0343	Byteposition in Bildschirm-speicherbyte
$0345	Lo-Byte der Bildschirmadresse
JSR$AEFD	prüft auf Komma
JSR$B7EB	holt X- und Y-Koordinaten vom BASIC
	XL - $14
	XH - $15
	Y - X-Register
JSR$C035	löscht HIRES-Bildschirm
JSR$0073	holt Byte (CHRGET)
JSR$0079	letztes Byte (CHRGOT)
JMP$A7E7	zur Interpreterschleife
JMP$A7AE	nächster BASIC-Befehl

```
10 FORI=49152TO49518:READA:POKEI,A:B=B+A:NEXT
20 IFB<>35299THENPRINT"DATA ERROR"
30 DATA 169,12,141,8,3,169,192,141
40 DATA 9,3,96,234,32,115,0,201
50 DATA 92,240,6,32,121,0,76,231
60 DATA 167,32,115,0,162,0,221,61
70 DATA 192,240,8,232,224,3,208,246
80 DATA 76,174,167,189,71,193,141,56
90 DATA 192,189,74,193,141,57,192,32
100 DATA 90,193,76,174,167,79,81,67
110 DATA 32,115,0,32,235,183,142,91
120 DATA 192,165,20,208,3,32,115,192
130 DATA 169,59,141,17,208,169,24,141
140 DATA 24,208,169,2,162,255,157,0
150 DATA 4,157,0,5,157,0,6,157
160 DATA 0,7,224,0,240,4,202,76
170 DATA 94,192,96,169,0,160,63,140
180 DATA 126,192,162,255,157,0,32,224
190 DATA 0,240,4,202,76,124,192,192
200 DATA 32,240,4,136,76,119,192,96
210 DATA 32,115,0,32,235,183,138,141
220 DATA 80,3,41,248,74,74,74,72
230 DATA 173,80,3,41,7,141,65,3
240 DATA 165,20,41,248,141,66,3,165
250 DATA 20,41,7,170,169,128,224,0
260 DATA 240,5,202,74,76,182,192,141
270 DATA 67,3,104,170,189,8,193,188
280 DATA 40,193,72,24,152,109,65,3
290 DATA 141,69,3,176,3,76,221,192
300 DATA 24,104,105,1,72,173,69,3
310 DATA 109,66,3,141,69,3,104,144
320 DATA 3,24,105,1,101,21,141,255
330 DATA 192,141,5,193,173,69,3,141
340 DATA 254,192,141,4,193,173,63,63
350 DATA 96,234,234,141,63,63,96,0
360 DATA 32,33,34,35,37,38,39,40
370 DATA 42,43,44,45,47,48,49,50
380 DATA 52,53,54,55,57,58,59,60
390 DATA 62,63,0,0,0,0,0,0
400 DATA 0,64,128,192,0,64,128,192
410 DATA 0,64,128,192,0,64,128,192
420 DATA 0,64,128,192,0,64,128,192
430 DATA 0,64,0,0,0,0,0,64
440 DATA 80,90,192,193,193,0,0,0
450 DATA 32,144,192,13,67,3,32,3
460 DATA 193,96,32,144,192,72,173,67
470 DATA 3,73,255,141,67,3,104,45
480 DATA 67,3,32,3,193,96,0,0

READY.
```

```
B*
   PC  SR AC XR YR SP
.;C400 30 60 79 FF F6
.
C000 A9 0C       LDA #$0C      Initialisierung
C002 8D 08 03    STA $0308
C005 A9 C0       LDA #$C0
C007 8D 09 03    STA $0309
C00A 60          RTS
C00B EA          NOP
C00C 20 73 00    JSR $0073     Vergleich mit Sonderzeichen "t"
C00F C9 5C       CMP #$5C
C011 F0 06       BEQ $C019
C013 20 79 00    JSR $0079
C016 4C E7 A7    JMP $A7E7
C019 20 73 00    JSR $0073
C01C A2 00       LDX #$00
C01E DD 3D C0    CMP $C03D,X   Welcher Befehl ?
C021 F0 08       BEQ $C02B
C023 E8          INX
C024 E0 03       CPX #$03
C026 D0 F6       BNE $C01E
C028 4C AE A7    JMP $A7AE     keiner, dann "Syntax Error"
C02B BD 47 C1    LDA $C147,X   Anfangsadresse aus Tabelle holen
C02E 8D 38 C0    STA $C038
C031 BD 4A C1    LDA $C14A,X
C034 8D 39 C0    STA $C039
C037 20 5A C1    JSR $C15A     Befehl ausführen
C03A 4C AE A7    JMP $A7AE     Zurück zur Interpreterschleife
C03D 4F          ???           Tabelle: O,Q,C
C03E 51 43       EOR ($43),Y
C040 20 73 00    JSR $0073     HIRES einschalten
C043 20 EB B7    JSR $B7EB
C046 8E 5B C0    STX $C05B
C049 A5 14       LDA $14
C04B D0 03       BNE $C050
C04D 20 73 C0    JSR $C073     HIRES CLEAR
C050 A9 3B       LDA #$3B
C052 8D 11 D0    STA $D011
C055 A9 18       LDA #$18
C057 8D 18 D0    STA $D018
C05A A9 02       LDA #$02
C05C A2 FF       LDX #$FF
C05E 9D 00 04    STA $0400,X
C061 9D 00 05    STA $0500,X
C064 9D 00 06    STA $0600,X
C067 9D 00 07    STA $0700,X
```

```
C06A E0 00       CPX #$00
C06C F0 04       BEQ $C072
C06E CA          DEX
C06F 4C 5E C0    JMP $C05E
C072 60          RTS
C073 A9 00       LDA #$00        Subroutine HIRES CLEAR
C075 A0 3F       LDY #$3F
C077 8C 7E C0    STY $C07E
C07A A2 FF       LDX #$FF
C07C 9D 00 20    STA $2000,X
C07F E0 00       CPX #$00
C081 F0 04       BEQ $C087
C083 CA          DEX
C084 4C 7C C0    JMP $C07C
C087 C0 20       CPY #$20
C089 F0 04       BEQ $C08F
C08B 88          DEY
C08C 4C 77 C0    JMP $C077
C08F 60          RTS
C090 20 73 00    JSR $0073       Bildschirm Position des Punktes
C093 20 EB B7    JSR $B7EB       berechnen
C096 8A          TXA
C097 8D 3C 03    STA $033C
C09A 29 F8       AND #$F8
C09C 4A          LSR
C09D 4A          LSR
C09E 4A          LSR
C09F 48          PHA
C0A0 AD 3C 03    LDA $033C
C0A3 29 07       AND #$07
C0A5 8D 41 03    STA $0341
C0A8 A5 14       LDA $14
C0AA 29 F8       AND #$F8
C0AC 8D 42 03    STA $0342
C0AF A5 14       LDA $14
C0B1 29 07       AND #$07
C0B3 AA          TAX
C0B4 A9 80       LDA #$80
C0B6 E0 00       CPX #$00
C0B8 F0 05       BEQ $C0BF
C0BA CA          DEX
C0BB 4A          LSR
C0BC 4C B6 C0    JMP $C0B6
```

```
.
C0BF 8D 43 03    STA $0343
C0C2 68          PLA
C0C3 AA          TAX
C0C4 BD 08 C1    LDA $C108,X
C0C7 BC 28 C1    LDY $C128,X
C0CA 48          PHA
C0CB 18          CLC
C0CC 98          TYA
C0CD 6D 41 03    ADC $0341
C0D0 8D 45 03    STA $0345
C0D3 B0 03       BCS $C0D8
C0D5 4C DD C0    JMP $C0DD
C0D8 18          CLC
C0D9 68          PLA
C0DA 69 01       ADC #$01
C0DC 48          PHA
C0DD AD 45 03    LDA $0345
C0E0 6D 42 03    ADC $0342
C0E3 8D 45 03    STA $0345
C0E6 68          PLA
C0E7 90 03       BCC $C0EC
C0E9 18          CLC
C0EA 69 01       ADC #$01
C0EC 65 15       ADC $15
C0EE 8D FF C0    STA $C0FF
C0F1 8D 05 C1    STA $C105
C0F4 AD 45 03    LDA $0345
C0F7 8D FE C0    STA $C0FE
C0FA 8D 04 C1    STA $C104
C0FD AD 3F 3F    LDA $3F3F
C100 60          RTS
C101 EA          NOP
C102 EA          NOP
C103 8D 3F 3F    STA $3F3F      Punkt setzen bzw. löschen
C106 60          RTS
.
.
C150 20 90 C0    JSR $C090      Punkt setzen
C153 0D 43 03    ORA $0343
C156 20 03 C1    JSR $C103
C159 60          RTS
C15A 20 90 C0    JSR $C090      Punkt löschen
C15D 48          PHA
C15E AD 43 03    LDA $0343
C161 49 FF       EOR #$FF
C163 8D 43 03    STA $0343
C166 68          PLA
C167 2D 43 03    AND $0343
C16A 20 03 C1    JSR $C103
C16D 60          RTS
.
```

```
.
.:C10B 23 25 26 27 28 2A 2B 2C   Multiplikations- und Adresstabelle
.:C113 2D 2F 30 31 32 34 35 36
.:C11B 37 39 3A 3B 3C 3E 3F 00
.:C123 00 00 00 00 00 00 40 80
.:C12B C0 00 40 80 C0 00 40 80
.:C133 C0 00 40 80 C0 00 40 80
.:C13B C0 00 40 80 C0 00 40 00
.:C143 00 00 00 00 40 50 5A C0
.:C14B C1 C1 00 00 00 20 90 C0
.
.
```

```
10000 PRINT"■":ONAAGOTO10170,10060,10390
10010 :
10020 REM**********
10030 REM* CIRCLE *
10040 REM**********
10050 :
10060 A=π/40:FORI=0TO2*πSTEPA
10070 X1=MX+R*SIN(I)
10080 Y1=MY+R*COS(I)
10090 X2=MX+R*SIN(I+A)
10100 Y2=MY+R*COS(I+A)
10110 GOSUB10170:NEXT:RETURN
10120 :
10130 REM********
10140 REM* LINE *
10150 REM********
10160 :
10170 IFX1>X2THENH=X1:X1=X2:X2=H:H=Y1:Y1=Y2:Y2=H
10180 IFABS(X1-X2)<.99THEN10310
10190 M=(Y2-Y1)/(X2-X1)
10200 IFABS(M)<1THEN10270
10210 FORX=X1TOX2
10220 YA=Y1+M*(X-X1)
10230 YB=Y1+M*(X+1-X1)
10240 IFYA>YBTHENH=YA:YA=YB:YB=H
10250 FORY=YATOYB:£QX,Y:NEXT
10260 NEXT:GOTO10300
10270 FORX=X1TOX2
10280 Y=Y1+M*(X-X1)
10290 £QX,Y:NEXT
10300 RETURN
10310 IFY1>Y2THENH=Y1:Y1=Y2:Y2=H
10320 FORY=Y1TOY2:£QX1,Y:NEXT
10330 RETURN
10340 :
10350 REM*******
10360 REM* BOX *
10370 REM*******
10380 :
10390 IFX1>X2THENH=X1:X1=X2:X2=H:H=Y1:Y1=Y2:Y2=H
10400 FORX=X1TOX2
10410 £QX,Y1:£QX,Y2:NEXT
10420 IFY1>Y2THENH=Y1:Y1=Y2:Y2=H
10430 FORY=Y1TOY2:£QX1,Y:£QX2,Y:NEXT
10440 RETURN

READY.
```

Eine Hilfestellung für wirtschaftliche Entscheidungen sind Programmsammlungen, die die guten Grafik- und Farbmöglichkeiten des Computers nutzen. Diagramme, Sprites, optische Darstellungen von Simulationen werden eingesetzt, die die Ergebnisse verdeutlichen. Die finanzmathematischen Grundlagen sind zu jedem Programm beschrieben.

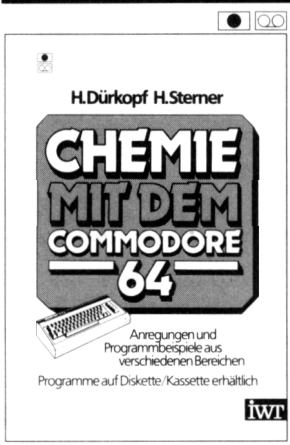
1983. 224 Seiten. Mit mehr. Abb. **Spiralh. DM 38,–/Fr. 38.–/S 342,–**
ISBN 3-88322-030-2

Bekanntlich verfügt der C 64 ›von Haus aus‹ über einen Baustein, der die Erzeugung von mehrstimmiger Musik erlaubt. Sowohl der Anfänger ohne musikalische Vorkenntnisse wird angesprochen, als auch der Musiker, der seine Ideen mit Hilfe des Computers umsetzen möchte.

In Vorb. Mai 1984. Ca. 200 Seiten. **Spiralh. Ca. DM 38,–/ca. Fr. 38.–/ca. S 342,–**
ISBN 3-88322-046-9

Dieses Buch führt nach einer Kurzbeschreibung der Grundlagen direkt in die Maschinensprachen-Programmierung ein, ohne erst die Befehle ›auf dem Trockenen‹ zu besprechen. Der Benutzer arbeitet sofort mit lauffähigen Programmen. Befehle werden dann eingeführt, wenn sie erforderlich sind.

In Vorb. April 1984. Ca. 260 Seiten. **Spiralh. Ca. DM 56,–/ca. Fr. 56.–/ca. S 498,–**
ISBN 3-88322-047-7



Programme reichen von stöchiometrischen Berechnungen bis zur Elementedatei. Besonders genutzt werden die grafischen Möglichkeiten. Die Anwendungsbeispiele gehen von Strukturformeln, grafischen Darstellungen von Versuchsanordnungen und Funktionen bis zur 3-D Grafik im Detail.

In Vorb. März 1984. Ca. 220 Seiten. **Spiralh. Ca. DM 48,–/ca. Fr. 48.–/ca. S 432,–**
ISBN 3-88322-049-3

Dieses Buch enthält eine ganze Reihe von sofort lauffähigen Spiel- und Simulationsprogrammen, möchte aber auch dazu anregen, diese Programme zu verändern und weiterzuentwickeln. Besonders reizvoll dürfte es wohl sein, den ›lernenden‹ Programmen noch etwas mehr ›Intelligenz‹ zu verleihen.

In Vorb. 1984. Ca. 200 Seiten. **Spiralh. Ca. DM 38,–/ca. Fr. 38.–/ca. S 342,–**
ISBN 3-88322-050-7

Grafikprogramme werden ›gehirngerecht‹ aufbereitet, d.h. man sieht, wie Grafikbefehle ›gehen‹: Neue Art des Formats – man bekommt ein ›Bild‹ des Befehls · Demo-Programme unterstützen das Gedächtnis · Bildschirm-Hardcopies als schnelles Nachschlagewerk · farbige Übersichtskarten zur Programmier-Erleichterung.

In Vorb. April 1984. Ca. 160 S. **Spiralh. Ca. DM 38,–/ca. Fr. 38.–/ca. S 342,–**
ISBN 3-88322-056-6

Der erste Band einer Reihe, die mit zahlreichen Programmen für Spiele und ›ernsthafte‹ Themen den Computer dem Benutzer näherbringt. Der Autor hat aus seiner Schulpraxis heraus Programme entwickelt, die das Lernen und Spielen mit dem Computer zum Vergnügen machen.

1983. 234 Seiten. **Kart. DM 32,–/Fr. 32.–/ S 288,–**
ISBN 3-88322-013-2

Jetzt wird es ernst: Hier wird Ihnen gezeigt, wie Sie mit dem Computer die Lohn- oder Einkommensteuererklärung erledigen, zeigt wohin die Staatsverschuldung geht – oder auch Ihre eigene, berechnet Ihre Zinsen (Soll oder Haben) auf der Bank, oder wie Sie Ihr Haus finanzieren können.

1983. 204 Seiten. **Kart. DM 32,–/Fr. 32.–/ S 288,–**
ISBN 3-88322-014-0

Dieses Buch enthält 40 mathematische Programme aus den Bereichen: Mehrregister-Arithmetik – Zahlentheorie – Kombinatorik – Algebra – Geometrie – numerische Mathematik. Neu ist die Langzahl-Arithmetik. Sie gestattet die Grundrechenarten für Zahlen bis 255 Stellen.

1983. 248 Seiten. **Kart. DM 42,–/Fr. 42.–/ S 378,–. ISBN 3-88322-016-7**

Der C 64 bietet vielseitige grafische Möglichkeiten. Dieses Buch gibt Informationen wie man Grafikfunktionen anwendet – Informationen, die man im Commodore-Handbuch nicht findet. Ausgehend von Grafiken mit den ›festen‹ Grafik-Zeichen wird systematisch zu den anspruchsvolleren Möglichkeiten, illustriert durch typische Beispiele, geführt.

1983. 138 S. 1 Folie. **Spiralh. DM 38,–/ Fr. 38.–/S 342,–. ISBN 3-88322-027-2**

Dieses Buch enthält 40 mathematische Programme aus den Bereichen: Mehrregister-Arithmetik – Zahlentheorie – Kombinatorik – Algebra – Geometrie – numerische Mathematik. Neu ist die Langzahl-Arithmetik. Sie gestattet die Grundrechenarten für Zahlen bis 255 Stellen.

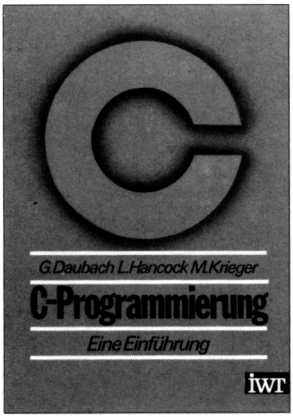
1984. 260 Seiten. **Kart. DM 42,–/ Fr. 42.–/S 378,–**
ISBN 3-88322-048-5

Dieses Buch bietet eine systematische Einführung in die Programmiersprache BASIC. Außer vielen kleineren Programmen zur Illustrierung der BASIC-Anweisungen gibt es eine umfangreiche Programmsammlung zu den verschiedensten Themenbereichen. Die besonderen Fähigkeiten des C 64 werden mit vielen Programmbeispielen erläutert.

1983. 356 Seiten. **Spiralh. DM 56,–/ Fr. 56.–/S 498,–**
ISBN 3-88322-029-9

Die Besonderheiten der 16-Bit Version von CBASIC werden erläutert. Spezielle Grafikbefehle sprechen das Betriebssystem GSX von DR direkt an. Weiter werden ›C‹-Funktionen mit CBASIC Programmen verknüpft, um so die Anwendung von Assembler zu umgehen. Vorteile von CP/M86, CCP/M86 und PC-DOS werden behandelt.

In Vorb. Juni 1984. Ca. 250 Seiten. **Spiralh. Ca. DM 68,–/ca. Fr. 68.–/ca. S 612,–. ISBN 3-88322-071-X**

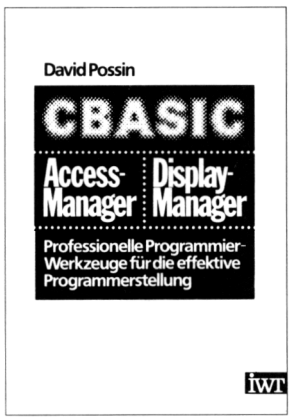


Die Programmiersprache ›C‹ ist besonders zur Erstellung schneller, maschinennaher Programme geeignet, weist jedoch gegenüber der Assemblerprogrammierung wesentliche Vorteile auf, wie z. B. lokale und globale Variable, Prozeduren, Funktionen usw. Diese Einführung setzt beim Leser keine umfassenden Kenntnisse voraus.

In Vorb. 1984. Ca. 250 Seiten. **Geb. ca. DM 56,–/ca. Fr. 56.–/ca. S 498,– ISBN 3-88322-041-8**

Wer hat nicht bereits verzweifelt versucht, das ›Computerchinesenglisch‹ zu verstehen? Hier hilft das Wörterbuch der Computerei mit seinen über tausend Begriffen. Außerdem sind die wichtigsten Begriffe erklärt. Ein handliches Nachschlagewerk für jeden, der sich mit Computerei beschäftigt.

1983. 2., erw. Aufl. 144 Seiten. **Kart. DM 32,–/Fr. 32.–/S 288,– ISBN 3-88322-026-4**



Zwei leistungsfähige Programme, die unter CP/M mit CBASIC professionell verknüpft werden: Access Manager für schnelle Dateiverwaltung, Display Manager als Maskengenerator. Diese und andere Einsatzmöglichkeiten werden beschrieben. Häufige Programmteile werden in Standardbibliotheken gespeichert und bei Bedarf neu geladen.

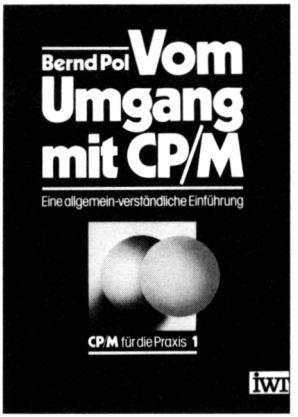
In Vorb. Mai 1984. Ca. 220 Seiten. **Spiralh. Ca. DM 58,–/ca. Fr. 58.–/ca. S 522,–. ISBN 3-88322-070-1**

Das Buch behandelt den Einsatz von Mikrocomputern im Fremdsprachenlernprozeß. Vom Inhalt: Vokabel-Trainingsprogramm, breiter Multiple-Choice Trainingsblock, Übungen zur Erweiterung des Wortschatzes, für Alltagssituationen etc. Zahlreiche Beispiele, engl. und franz., ausführlich kommentierte Programme.

In Vorb. April 1984. Ca. 220 Seiten. **Geb. Ca. DM 58,–/ca. Fr. 58.–/ca. S 522,– ISBN 3-88322-055-8**

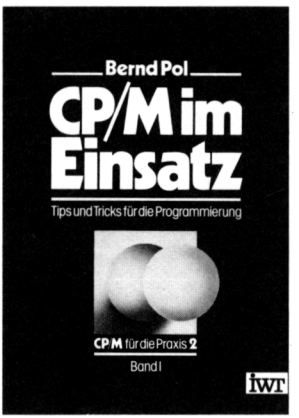
Das Buch ist sowohl für Programmierer als auch für Fans. Sie bekommen Einblick in das Zusammenwirken von zwei 6502-Prozessoren und den Datenverkehr über PIAs mit dem Rechner, dieses System arbeitet selbständig parallel. Ein Leckerbissen, wenn man den Computer noch besser nutzen will.

1982. 120 Seiten. Mit zahlr. Programmen. **RingO. DM 104,–/Fr. 104.–/S 936,– ISBN 3-88322-015-9**



Das Buch ist in drei Teile gegliedert: Teil 1 führt in die Eigenschaften von Computern und CP/M im besonderen ein. Aufbauend auf diesen Kenntnissen werden im 2. Teil die zentralen CP/M-Hilfsprogramme vorgestellt. Der 3. Teil geht – nach einer Einführung in die Funktionsweise des 8080-Prozessors – auf die CP/M-Systembesonderheiten ein.

1982. 386 S. Mit zahlr. prakt. Beispielen.
Geb. DM 48,–/Fr. 48.–/S. 432,–
ISBN 3-88322-004-3

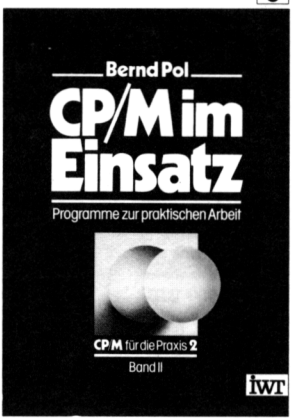


Dieser Band beschreibt wichtige Details des BDOS-Kerns und der CBIOS-Schnittstelle sowie Hinweise zur Fehlerverhütung. Weiter: Aufbau eines CBIOS-Systems, Fehlerbehandlung – Erweiterungsmöglichkeiten – Kompatibilitätsfragen zu MP/M und CP/Mplus.

In Vorb. 1984. Ca. 300 Seiten. Mit zahlreichen praktischen Beispielen.
Geb. Ca. DM 56,–/ca. Fr. 56.–/ca. S 498,–
ISBN 3-88322-006-X

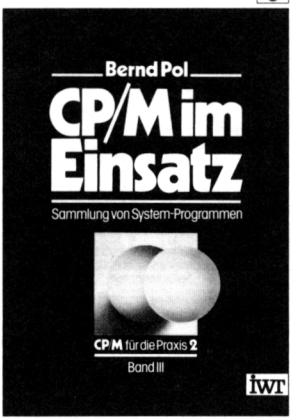
Dieses Buch beschreibt an Beispielen aus der Praxis das Arbeiten mit dem wohl meistbenutzten Datenbanksystem der Welt. Es werden keine Vorkenntnisse vorausgesetzt, sondern es wird Schritt für Schritt das komplette Wissen vermittelt, das man zum Arbeiten mit einer Datenbank braucht.

In Vorb. Febr. 1984. Ca. 240 Seiten.
Geb. DM 56,–/Fr. 56.–/S 498,–
ISBN 3-88322-038-8



Inhalt: Macrobibliothek zur strukturierten ASM-Progr., Standard-Prozeduren, System-Schnittst. zum Abfangen aller BDOS-Fehler unter CP/M 2.2, ähnlich CP/M+, MP/M II. Standardroutinen zur zeichenorientierten I/O. Programme in strukturierter Zwischensprache setzen relozierenden Makro-ASM RMAC von D.R. voraus.

In Vorb. 1984. Ca. 360 S. Mit zahlr. ausgearb. Progr. **Geb. Ca. DM 56,–/ca. Fr. 56.–/S 498.–. ISBN 3-88322-032-9**



Zentrale Programme für CP/M, Programme zur Nutzung der BDOS-Eigenschaften, Zentralen Datenzugriff, CBIOS-Schnittstelle, Dienstprogramme zur Erweiterung und direkten Diskettenzugriff, Routinen für Dateien. Programme kommentiert in strukturierter Zwischensprache. Relozierender Makro-ASM RMAC von DR nötig.

In Vorb. 1984. 350 Seiten. **Geb. Ca. DM 56,–/ca. Fr. 56.–/ca. S 498,–.**
ISBN 3-88322-033-7

Das Buch, auch als Einstieg in das Programmieren geeignet, zeigt dBase II als mächtige Programmiersprache für die Datenbankanwendung. Inhalt: Was ist eine Datenbank? Warum dBase II? Wie wird programmiert? Einführung: Autokostenverwaltung; für Fortgeschrittene: Literaturverwaltung, kommentierte Listings, Tips und Tricks.

In Vorb. April 1984. Ca. 240 Seiten. **Geb. Ca. DM 56,–/ca. Fr. 56.–/ca. S 498,–**
ISBN 3-88322-039-6

Dieses Buch führt Sie zur sicheren Handhabung der Tabellenkalkulation. Zahlreiche Anwendungen mit ausführlicher Beschreibung schließen sich an. Unter anderem: Werbeplanung, Reisekosten, Kapazitätsauslastung, Baufinanzierung, Reaktionszeiten.– Auch für die englische Version nutzbar.

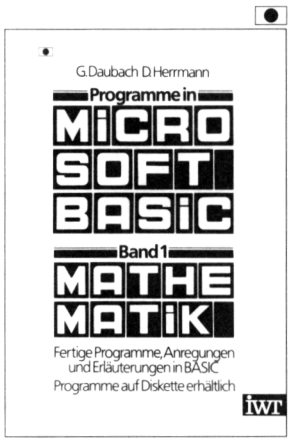
In Vorb. Mai 1984. Ca. 250 Seiten. **Spiralh. Ca. DM 56,–/ca. Fr. 56.–/ca. S 498,– ISBN 3-88322-074-4**

Dieses Buch wendet sich an SuperCalc-Anwender: Anfänger führt es schnell zur sicheren Handhabung, Fortgeschrittene finden ausgereifte Anwendungen. Beispiele mit ausführlicher Beschreibung machen das Buch auch für VisiCalc-Anwender interessant. Die Unterschiede zwischen VisiCalc und SuperCalc sind in einem Kapitel dargestellt.

1984. 248 Seiten. **Spiralh. DM 56,–/ Fr. 56.–/S 498,– ISBN 3-88322-040-X**

Die Verbindung von Kalkulation, Grafik und Datenbank verhalf Lotus zu schnellem Erfolg. In diesem Buch erlernen Sie zunächst die sichere Nutzung von Kalkulation und Grafik. Zahlreiche Beispiele zeigen die Leistungsfähigkeit von 1-2-3: Umsatzprognose, Investitionsrechnung, Tilgungsplan, Private Ausgaben u.a.

In Vorb. April 1984. Ca. 250 Seiten. **Geb. Ca. DM 58,–/ca. Fr. 58.–/ca. S 522,– ISBN 3-88322-085-X**

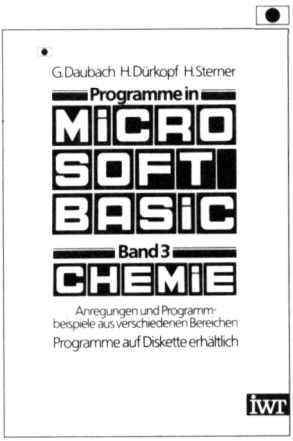


Dieses Buch enthält 40 mathematische Programme aus den Bereichen: Mehrregister-Arithmetik – Zahlentheorie – Kombinatorik – Algebra – Geometrie – numerische Mathematik. Neu ist die Langzahl-Arithmetik. Sie gestattet die Grundrechenarten für Zahlen bis 255 Stellen.

In Vorb. März 1984. 250 Seiten. **Spiralh. Ca. DM 48,–/ca. Fr. 48.–/ca. S 432,– ISBN 3-88322-077-9**

Eine Hilfestellung für wirtschaftliche Entscheidungen sind Programmsammlungen, die die guten Grafik- und Farbmöglichkeiten des Computers nutzen. Diagramme, Sprites, optische Darstellungen von Simulationen werden eingesetzt, die die Ergebnisse verdeutlichen. Die finanzmathematischen Grundlagen sind zu jedem Programm beschrieben.

In Vorb. März 1984. Ca. 200 Seiten. **Spiralh. Ca. DM 48,–/ca. Fr. 48.–/ca. S 432,– ISBN 3-88322-076-0**



Programme reichen von stöchiometrischen Berechnungen bis zur Elementedatei. Besonders genutzt werden die grafischen Möglichkeiten. Die Anwendungsbeispiele gehen von Strukturformeln, grafischen Darstellungen von Versuchsanordnungen und Funktionen bis zur 3-D Grafik im Detail.

In Vorb. Mai 1984. Ca. 220 Seiten. **Spiralh. Ca. DM 56,–/ca. Fr. 56.–/ca. S 498,– ISBN 3-88322-078-7**

Eine Hilfestellung für wirtschaftliche Entscheidungen sind Programmsammlungen, die die guten Grafik- und Farbmöglichkeiten des Computers nutzen. Diagramme, Sprites, optische Darstellungen von Simulationen werden eingesetzt, die die Ergebnisse verdeutlichen. Die finanzmathematischen Grundlagen sind zu jedem Programm beschrieben.

In Vorb. April 1984. Ca. 200 Seiten. **Spiralh. Ca. DM 48,–/ca. Fr. 48.–/ca. S 432,– ISBN 3-88322-060-4**

Dieses Buch enthält 40 mathematische Programme aus den Bereichen: Mehrregister-Arithmetik – Zahlentheorie – Kombinatorik – Algebra – Geometrie – numerische Mathematik. Neu ist die Langzahl-Arithmetik. Sie gestattet die Grundrechenarten für Zahlen bis 255 Stellen.

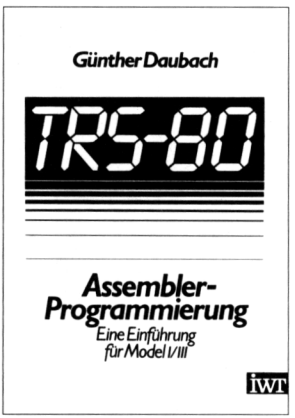
In Vorb. April 1984. 250 Seiten. **Spiralh. Ca. DM 56,–/ca. Fr. 56.–/ca. S 498,– ISBN 3-88322-061-2**

Programme reichen von stöchiometrischen Berechnungen bis zur Elementedatei. Besonders genutzt werden die grafischen Möglichkeiten. Die Anwendungsbeispiele gehen von Strukturformeln, grafischen Darstellungen von Versuchsanordnungen und Funktionen bis zur 3-D Grafik im Detail.

In Vorb. April 1984. Ca. 220 Seiten. **Spiralh. Ca. DM 56,–/ca. Fr. 56.–/ca. S 498,– ISBN 3-88322-062-0**

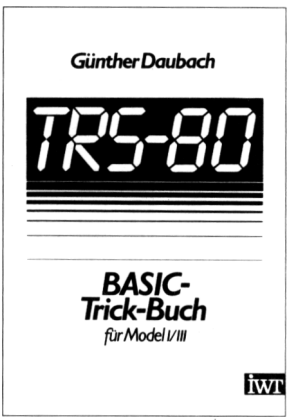
Die Programme sind in TI-BASIC geschrieben, verwenden die Grundkonfiguration des Computers und machen ausgiebig Gebrauch von den Farbgrafikmöglichkeiten sowie dem Tongenerator. Sie sind so angelegt, daß der Einsteiger schnell Erfolge erzielt, der Fortgeschrittene aber die Spielprogramme nach Belieben ergänzen, erweitern oder variieren kann.

1983. 220 Seiten. **Kart. DM 38,–/Fr. 38.–/S 342,–. ISBN 3-88322-045-0**



Assembler-Programmierung mit Z80, dadurch breit einsetzbar. Nach Einführung eine Erklärung des Befehlssatzes mit Beispielen als Bausteine für eigene Programme. Weiteres: Ein- und Ausgabe über Tastatur und Bildschirm, Verwendung ›eingebauter Routinen‹, eines Editors/-Assemblers. Hilfreiches Nachschlagewerk.

In Vorb. April 1984. Ca. 250 Seiten. **Kart. Ca. DM 52,–/ca. Fr. 52.–/ca. S 468,– ISBN 3-88322-017-5**



Für alle gedacht, die ihren TRS 80 besser nutzen wollen. Anfänger wie ›alte Hasen‹ finden eine Vielzahl nützlicher Tips und Tricks. Schwerpunkte: wie Basic-Programme schneller werden; wie man Basic-Programme strukturiert; die Verbindung zwischen Basic- und Maschinenprogrammen, universelle Masken u.a.m.

In Vorb. April 1984. Ca. 300 Seiten. **Kart. Ca. DM 52,–/ca. Fr. 52.–/ca. S 468,– ISBN 3-88322-018-3**

M. Weber

3-D-Grafik in Theorie und Praxis

Hintergründe der Programmierung am Beispiel Commodore 64

Nach einer kurzen Einführung in die Vektorrechnung werden, ausgehend von der Darstellung geometrischer Grundoperationen und der Erstellung einfacher Graphen, wie z. B. von Kreisen oder Ellipsen, die in beliebiger Lage auf den Bildschirm geplottet werden, die Darstellung von Kugeln und räumlichen Funktionen anhand einfacher Beispiele besprochen.
Es wird gezeigt, wie sich selbst komplizierte Operationen sehr einfach beschreiben lassen. Es wird demonstriert, wie sich dreidimensionale Probleme lösen lassen. Die Anwendungen reichen von der Darstellung einer Geraden über die Simulation des Dreikörperproblems bis hin zur dreidimensionalen Simulation eines Planetensystems.
Um den Leser nicht mit langwierigen BASIC-Routinen zur Ansteuerung der hochaufgelösten Grafik aufzuhalten, findet sich auch ein Maschinenprogramm, das in Verbindung mit einem BASIC-Unterprogramm die wichtigsten Steuerfunktionen, wie etwa Grafikbildschirm einschalten, Punkte setzen, sowie Kreise, Linien oder Rahmen ziehen, wesentlich beschleunigt. Das Maschinenprogramm wird dabei durch ein BASIC-Programm eingelesen, was es jedermann ermöglicht, davon Gebrauch zu machen.
Programme sind auch auf Diskette erhältlich.

ISBN 3-88322-052-3